Section 3: Specifying Guidance Variables

The CICS Component must evaluate CICS regions generated with a wide variety of CICS configuration options, use data from several different sources, and evaluate the constraints to improved CICS performance from the perspective of many different management objectives. Guidance variables are provided to the CICS Component to allow it to respond to the different configurations, different data sources, and different management objectives.

CPEXPERT.USOURCE(CICGUIDE) contains variables to establish the overall guidance for the CICS Component. You specify certain guidance variables when you initially use the CICS Component to evaluate a CICS region, and you modify the variables in the CICGUIDE member whenever you wish to change the guidance to CPExpert.

The variables in the CICGUIDE module can be viewed as "data selection and presentation" variables and "analysis guidance" variables. These two types of variables are discussed separately.

- The data selection and presentation variables allow you to select particular time intervals to be analyzed, specify the VTAM application identifier associated with the CICS region to be analyzed, suppress certain listings, etc. Chapter 1 describes the data selection and presentation variables and how they are used.

- The analysis guidance variables allow you to provide guidance to the CICS Component with regard to analysis thresholds that may be unique to your environment. The defaults for the various thresholds may be appropriate for the analysis performed by the CICS Component. However, you may have unique situations (or you may simply disagree with the defaults selected). Chapter 3 describes the analysis guidance variables and their defaults.

  The analysis guidance variables were designed to allow you to tailor the analysis performed by CPExpert. **Do not hesitate to make changes if the defaults for the analysis guidance variables do not meet your needs**.

Please do not allow CPExpert to perform analysis or produce reports which are meaningless in your environment. If the analysis and reports produced by CPExpert do not meet your needs, alter the guidance to CPExpert. If the guidance is insufficient, please call Computer Management Sciences at (703) 922-7027 (or e-mail **Don_Deese@cpexpert.com**) so we can make changes to improve CPExpert for you!

Chapter 1: Data Selection and Presentation Variables

The data selection and presentation variables allow you to select particular data to be analyzed and how the results are to be presented. This chapter describes these variables and how the variables are used.

Exhibit 3-1 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the data selection and presentation variables.

For most users, the data selection variables rarely will be modified. This is because the CICS Component normally will process a day's data, and the data will be processed after the data has been placed into a performance data base. The daily files will contain the data you wish CPExpert to evaluate.

The data selection variables allow you to identify the data that the CICS Component is to analyze, even though the CICS data might be in a different performance data base than your standard performance data base (for example, your standard data base might be created by MICS, while the CICS interval statistics might be created by MXG). This capability to identify the location of CICS statistics extends down to the individual SAS file containing CICS statistics.

Some users might wish to restrict the analysis to only a specific shift (for example, you might not particularly care about CICS performance at times other than the prime shift). The data selection variables may be used to select specific measurement intervals to analyze (for example, you may notice that CICS performance is particularly poor during some interval and wish analysis focused only on that interval).

The data presentation variables might be modified after you initially execute the CICS Component, but these variables will not normally be modified after you have established the reports to your satisfaction.


```
*************************************************************************;
*             DATA SELECTION AND PRESENTATION VARIABLES                 ;
*************************************************************************;
%LET CICLIB  = &PDBLIB ; * SAS LIBRARY CONTAINING CICS DATA             ;
%LET STANDARD= YES     ; * STANDARD MXG DATA SETS ARE USED              ;
%LET CICAUTO = &CICLIB..CICAUTO ; * SPECIFY MXG CICAUTO  SAS LIB.FILE  ;
%LET CICCONMR= &CICLIB..CICCONMR; * SPECIFY MXG CICCONMR SAS LIB.FILE  ;
%LET CICCONSR= &CICLIB..CICCONSR; * SPECIFY MXG CICCONSR SAS LIB.FILE  ;
%LET CICCONSS= &CICLIB..CICCONSS; * SPECIFY MXG CICCONSS SAS LIB.FILE  ;
.
.
.
.
%LET CICLGS  = &CICLIB..CICLGS  ; * SPECIFY MXG CICLGS   SAS LIB.FILE  ;
%LET CICLSRFR= &CICLIB..CICLSRFR; * SPECIFY MXG CICLSRFR SAS LIB.FILE  ;
%LET CICLSRR = &CICLIB..CICLSRR ; * SPECIFY MXG CICLSRR  SAS LIB.FILE  ;
%LET CICNQG  = &CICLIB..CICNQG  ; * SPECIFY MXG CICNQG   SAS LIBRARY   ;
%LET CICRMG  = &CICLIB..CICRMG  ; * SPECIFY MXG CICRMG   SAS LIBRARY   ;
%LET CICSDG  = &CICLIB..CICSDG  ; * SPECIFY MXG CICSDG   SAS LIB.FILE  ;
%LET CICSMDSA= &CICLIB..CICSMDSA; * SPECIFY MXG CICSMDSA SAS LIB.FILE  ;
%LET CICTC   = &CICLIB..CICTC   ; * SPECIFY MXG CICTC    SAS LIB.FILE  ;
%LET CICTCR  = &CICLIB..CICTCR  ; * SPECIFY MXG CICTCR   SAS LIB.FILE  ;
%LET CICTCLR = &CICLIB..CICTCLR ; * SPECIFY MXG CICTCLR  SAS LIB.FILE  ;
%LET CICTSQ  = &CICLIB..CICTSQ  ; * SPECIFY MXG CICTSQ   SAS LIB.FILE  ;
%LET CICTSR  = &CICLIB..CICTSR  ; * SPECIFY MXG CICTSR   SAS LIB.FILE  ;
%LET CICVT   = &CICLIB..CICVT   ; * SPECIFY MXG CICVT    SAS LIB.FILE  ;
%LET CICXMC  = &CICLIB..CICXMC  ; * SPECIFY MXG CICXMC   SAS LIB.FILE  ;
%LET CICXMG  = &CICLIB..CICXMG  ; * SPECIFY MXG CICXMG   SAS LIB.FILE  ;
%LET CICXMR  = &CICLIB..CICXMR  ; * SPECIFY MXG CICXMR   SAS LIB.FILE  ;
%LET CICXQ1  = &CICLIB,,CICXQ1  ; * SPECIFY MXG CICXQ1   SAS LIBRARY   ;
%LET CICXQ2  = &CICLIB..CICXQ2  ; * SPECIFY MXG CICXQ2   SAS LIBRARY   ;
%LET CICXQ3  = &CICLIB..CICXQ3  ; * SPECIFY MXG CICXQ3   SAS LIBRARY   ;
%LET EXTRACT = N                ; * EXTRACT CICS FILES/VARIABLES       ;
```

SAMPLE DISPLAY OF CPEXPERT.USOURCE(CICGUIDE) MODULE

EXHIBIT 3-1


```
%LET CICDATES=01FEB1991; * START DATE FOR DATA ANALYSIS                ;
%LET CICTIMES=08:00:00 ; * START TIME FOR DATA ANALYSIS                ;
%LET CICDATEE=31DEC9999; * END DATE FOR DATA ANALYSIS                  ;
%LET CICTIMEE=16:00:00 ; * END TIME FOR DATA ANALYSIS                  ;
%LET CICDAT2S= 0       ; * DEFAULT SECOND SELECTION DATE - START       ;
%LET CICTIM2S= 0       ; * DEFAULT SECOND SELECTION TIME - START       ;
%LET CICDAT2E= 0       ; * DEFAULT SECOND SELECTION DATE - END         ;
%LET CICTIM2E= 0       ; * DEFAULT SECOND SELECTION TIME - END         ;
%LET SHIFT   = N       ; * START AND END TIMES REFER TO SHIFT          ;
%LET SYSTEM  =*ALL     ; * PROCESS ALL SYSTEMS                         ;
%LET SYSTEMn =system   ; * PROCESS SYSTEMn (n = 1-9)                   ;
%LET ALL_CICS= N       ; * OPTION TO ANALYZE ALL CICS REGIONS          ;
%LET APPLID1 =         ; * APPLID OF THE CICS REGION TO ANALYZE        ;
%LET APPLIDn =         ; * APPLID OF THE CICS REGION TO ANALYZE        ;
%LET MAXAPPL = 99      ; * MAXIMUM CICS REGIONS TO ANALYZE             ;
%LET LISTOFF = Y       ; * OPTION: LIST RULES THAT WERE TURNED OFF     ;
%LET LISTGDE = Y       ; * OPTION: LIST GUIDANCE PROVIDED TO CPEXPERT  ;
%LET CICSUMRY = N      ; * PRODUCE SUMMARY REPORT                      ;
%LET CICS_SYS = N      ; * PRODUCE SUMMARY REPORT BY SYSTEM            ;
%LET SASODS  = N       ; * CONTROLS WHETHER SAS ODS IS USED            ;
%LET PATH    =         ; * PATH FOR ODS OUTPUT                        ;
%LET FRAME   = CICFRAME; * GENERIC ODS FRAME NAME                      ;
%LET CONTENTS= CICSCONT; * GENERIC ODS CONTENTS NAME                   ;
%LET BODY    = CICSBODY; * GENERIC ODS BODY NAME                      ;
%LET LINKPDF =         ; * INSERT HTML LINKS TO PDF FILES             ;
%LET STYLE   =         ; * ODS HTML STYLE OPTION                      ;
%LET PDFODS  = N       ; * CONTROLS WHETHER SAS PDF IS USED           ;
%LET PDFFILE = filename; * DEFINES THE SAS PDF OUTPUT FILE             ;
%LET URL     = N       ; * CONTROLS .HTM IN SAS ODS FRAME OUTPUT      ;
%LET SYS_FRAM=         ; * USE GENERIC ODS FRAME NAME                 ;
%LET SYS_CONT=         ; * USE GENERIC ODS CONTENTS NAME              ;
%LET SYS_BODY=         ; * USE GENERIC ODS BODY NAME                  ;
%LET SYS_PATH=         ; * USE GENERIC ODS PATH                       ;
```

SAMPLE DISPLAY OF CPEXPERT.USOURCE(CICGUIDE) MODULE

EXHIBIT 3-1 (Continued)

Chapter 1.1: Specifying SAS library containing CICS data

The optional **CICLIB** guidance variable allows you to specify a SAS library containing your CICS performance data. This SAS library can be different from the SAS library containing your normal performance data acquired from SMF, RMF, etc. The default value for the CICLIB guidance variable points to the SAS library identified by the PDBLIB guidance variable in USOURCE(GENGUIDE).

You should alter the default CICLIB guidance variable only if your CICS performance data base is contained in a SAS library different from the normal performance data base. The value specified for CICLIB is used by CPExpert as the DDNAME used to access CICS performance data.

Chapter 1.2: Specifying standard MXG CICS data sets

The optional **STANDARD** guidance variable applies only if you have tailored the variables retained in the CICS files in your MXG performance data base, or if you do not retain all files specified in Exhibit 3-1. Most organizations using MXG process CICS interval statistics using the standard MXG software. Some organizations tailor the software (using standard MXG exits or modifying the code), to drop variables from the CICS data sets created by MXG or to suppress MXG files.

If you have **not** modified MXG to tailor the CICS data sets or to suppress MXG files, you should specify **%LET STANDARD = YES;** to advise CPExpert that MXG has not been modified. CPExpert will not generate the code to determine which CICS data sets and which variables are present in your MXG performance data base (thus saving processing time). The default value for the **STANDARD** variable is "YES", since most organizations do not modify the MXG data collection process.

If you **have** modified MXG to tailor the CICS data sets, you should specify **%LET STANDARD = NO;** to advise CPExpert that MXG has been modified. CPExpert will generate the code to determine which CICS data sets and which variables are present in your MXG performance data base. CPExpert will suppress any rules that depend upon missing data (and CPExpert will advise you that the rules have been suppressed).

Chapter 1.3: Specifying library.file for MXG data sets

The optional **CICAUTO, CICCFS6D, CICCFS7D, CICCFS8D, CICCFS9D, CICCONMR, CICCONSR, CICCONSS, CICDB2GL, CICDB2RE, CICDLIG, CICDLIT, CICDQG, CICDS, CICDTB, CICFCR, CICJCR, CICLDR, CICLGS, CICLSRFR, CICLSRR, CICNQG, CICRMG, CICSDG, CICSMDSA, CICSMT, CICTC, CICTCR, CICTCLR, CICTSQ, CICTSR**, **CICVT**, **CICXMC**, **CICXMG**, **CICXMR, CICXQ1, CICXQ2,** and **CICXQ3** guidance variables allow you to specify a SAS library.file for any MXG CICS data set used by CPExpert.

These SAS libraries.files can be different from the SAS library.file containing any other MXG data.

The default values for these optional MXG guidance variables point to the SAS library identified by the CICLIB guidance variable in USOURCE(CICGUIDE), which in turn normally points to the SAS library identified by the PDBLIB guidance variable in USOURCE(GENGUIDE). You should alter the default guidance variables only if you have any of these CICS files contained in a SAS library different from the normal CICS performance data base. Additionally, you can specify a file name different from the standard MXG file.

Chapter 1.4: EXTRACT variable

Some CPExpert users have an extremely large CICS environment, and the CICS performance staff is unable to allocate easily the DASD space for SAS libraries because of the significant amount of space required. The **EXTRACT** variable was introduced so these users could extract only the APPLIDs that they wished to analyze and to retain only those variables required by the CICS Component.

When **%LET EXTRACT=Y;** is specified, CPExpert extracts only the MXG data sets and only the MXG variables that are necessary for the CICS Component to analyze CICS performance constraints. **Only the APPLIDs specified (as described in the Chapter 1.10: APPLIDn variable) will be selected.**

The **CICEXTR** module in CPEXPERT.SOURCE contains sample code that can be used to extract SAS data sets and variables.

Chapter 1.5: CICDATES and CICTIMES variables

The CICDATES and CICTIMES variables specify the start date and start time, respectively, for the interval the CICS Component is to analyze. These variables (in conjunction with the CICDATEE and CICTIMEE variables) allow you to select specific periods of data to analyze. For example, to specify that data selection should start at 08:00:00 on March 4, 2002, specify:

```
%LET CICDATES = 04MAR2002;  * START DATE FOR DATA ANALYSIS;
%LET CICTIMES = 08:00:00;   * START TIME FOR DATA ANALYSIS;
```


The CICDATES and CICTIMES variables (and the CICDATEE and CICTIMEE variables described below) are not normally altered from their defaults. These variables are used **only** if you wish to analyze a subset of the data contained in your performance data base. Under most conditions, you simply use the defaults and CPExpert will analyze all available CICS statistics.

Chapter 1.6: CICDATEE and CICTIMEE variables

The CICDATEE and CICTIMEE variables specify the end date and end time, respectively, for the interval of SMF data the CICS Component is to analyze. For example, to specify that data selection should end at 17:00:00 on March 8, 2002, specify:

```
%LET CICDATEE = 08MAR2002;  * END DATE FOR DATA ANALYSIS;
%LET CICTIMEE = 17:00:00;   * END TIME FOR DATA ANALYSIS;
```

Chapter 1.7: CICDAT2S and CICTIM2S variables

The CICDAT2S and CICTIM2S variables are optional. These variables specify the start date and start time, respectively, for a second interval of CICS statistics that the CICS Component is to analyze. These variables (in conjunction with the optional CICDAT2E and CICTIM2E variables) allow you to select a second period of data to analyze, in addition to the period specified by the CICDATES/CICTIMES and CICDATEE/CICTIMEE selection variables. For example, to specify that a second period of data selection should start at 20:00:00 on March 4, 2002, specify:

```
%LET CICDAT2S = 04MAR2002; * START DATE FOR DATA ANALYSIS;
%LET CICTIM2S = 20:00:00;  * START TIME FOR DATA ANALYSIS;
```

Chapter 1.8: CICDAT2E and CICTIM2E variables

The CICDAT2E and CICTIM2E variables are optional. These variables specify the end date and end time, respectively, for a second interval of CICS statistics the CICS Component is to analyze. These variables (in conjunction with the optional CICDAT2S and CICTIM2S variables) allow you to select a second period of data to analyze, in addition to the period specified by the CICDATES/CICTIMES and CICDATEE/CICTIMEE selection variables. For example, to specify that a second period of data selection should end at 22:00:00 on March 4, 2002, specify:


```
%LET CICDAT2E = 08MAR2002;     * END DATE FOR DATA ANALYSIS;
%LET CICTIM2E = 22:00:00;      * END TIME FOR DATA ANALYSIS;
```

Chapter 1.9: SHIFT variable

The SHIFT variable is used with the CICDATES, CICTIMES, CICDATEE, and CICTIMEE variables. CPExpert will use the CICDATES and CICTIMES variables to exclude data before the specified date and time. The SHIFT variable allows you indicate how the time-selection variables should be used.

- If the SHIFT variable is "N", the time-selection will be based upon the absolute start and end dates/times specified. For example, if you wish CPExpert to process **all** data during a week, the start date and start time would be specified as the beginning of the week, and the end date and end time would be specified as the end of the week. You would specify "%LET SHIFT = N;" to process each 24-hour day. In the example shown above, data would be processed from 08:00:00 on 4 March until 17:00:00 on 8 March.

- If the SHIFT variable is "Y", the time-selection will be based upon the start and end dates, and the start and end times within each selected date. In the example shown above, perhaps you wished to process only the daily shift beginning at 08:00:00 and ending at 17:00:00. You would specify "%LET SHIFT = Y;" to process only the identified shift data, during the selected dates.

Chapter 1.10: SYSTEM variable

The SYSTEM variable is used to specify whether all systems in the performance data base should be evaluated, or to select a specific system identification to be evaluated.

Some users have data from multiple systems in their performance data base. For many of these users, or for users who have data for a single system represented in their performance data base, the default "*ALL" will be appropriate. No change of the SYSTEM variable would be required for these users.

However, some users who have data from multiple systems in their performance data base may wish to evaluate only a single system with the parameters specified in this member of CICGUIDE. For example, they might be temporarily interested in evaluating the performance of only an "important" system (such as a major production system) and not be interested in evaluating the performance of other systems with data in the performance data base. This evaluation can be accomplished by changing the SYSTEM variable to

specify the system identification to be evaluated. For example, to specify that only data from SYS1 should be evaluated, specify:

```
%LET SYSTEM = SYS1 ; * PROCESS ONLY DATA FROM SYS1;
```

In another situation, a CPExpert user might wish to evaluate different systems with different CICGUIDE parameters. These different evaluations can be accomplished by different executions of the CICS Component[1]. For each execution of the CICS Component, the USOURCE DD statement would be changed to reference different USOURCE libraries. Each USOURCE library would contain guidance members with appropriate guidance variables. The SYSTEM variable for each CICGUIDE guidance member would specify the system identification to which the guidance applied.

Please note that if the **ALL_CICS** guidance variable (described below) is specified as %LET ALL_CICS=Y; to specify analysis of all APPLIDs, the SYSTEM specification **takes precedence** (that is, if SYSTEM is specified as %LET SYSTEM=ssss, only APPLIDs for the "ssss" system will be analyzed.

Chapter 1.11: SYSTEMn variable(s)

The SYSTEMn variable(s) are used to select multiple systems to be evaluated.

As described in the SYSTEM guidance variable discussion above, some sites have data from multiple systems in their performance data base. These sites can process data from all systems by specifying %LET SYSTEM=ALL; in USOURCE(CICGUIDE), or can select a specific system to process by specifying %LET SYSTEM=system; in USOURCE(CICGUIDE), where "system" is the system identification of the system to be processed.

Some sites have data from multiple systems in their performance data base and do not want to process all systems, but do wish to process more than one system. For example, some systems might be production systems and some might be test systems. For these sites, the **SYSTEMn** guidance variable can be used to select more than one specific system to analyze.

The SYSTEM guidance variable can be used one system to analyze, and the SYSTEMn guidance variable(s) can be used to select up to 9 additional systems to analyze. For

[1] A much better approach might be to specify unique guidance for the CICS regions involved, using the APPLID override guidance options described later in this section. The APPLID override guidance allows unique guidance to be specified for any CICS region.

example, if you wish to analyze data from four systems (named SYSA, SYSB, SYSC, AND SYSX) in a single execution of the CICS Component, specify:

```
%LET SYSTEM  = SYSA ; * PROCESS DATA FROM SYSA;
%LET SYSTEM1 = SYSB ; * PROCESS DATA FROM SYSB;
%LET SYSTEM2 = SYSC ; * PROCESS DATA FROM SYSC;
%LET SYSTEM3 = SYSX ; * PROCESS DATA FROM SYSX;
```

Chapter 1.12: Analyzing all CICS regions in PDB (ALL_CICS variable)

The default approach to analyzing CICS performance is to analyze a single CICS region, or to analyze multiple CICS regions **only if you specifically describe the name of each region (using the APPLIDn approach described below)**. This design is the default because analysts often wish to review the performance only of one or more important CICS regions.

The default approach is unwieldy with large CICS environments in which many CICS regions exist, since an analyst must spend an inordinate amount of time preparing guidance information for CPExpert. With a performance data base that may include hundreds of cloned CICS regions, the default approach becomes unworkable. The **ALL_CICS** guidance variable directs the CICS Component to analyze all CICS regions present in a performance data base.

- If the ALL_CICS guidance variable is specified as **%LET ALL_CICS = Y**, CPExpert will analyze all CICS regions with interval statistics contained in the performance data base.

- The default for the ALL_CICS guidance variable is **%LET ALL_CICS = N** (which means that CPExpert should analyze only those regions specified by the APPLIDn guidance variables described below). If this default is used, each region to be analyzed must be identified to the CICS Component of CPExpert, using USOURCE(CICGUIDE), using the APPLIDn approach described below.

Chapter 1.13: CICS region(s) to analyze - APPLIDn variable

The APPLIDn variable (where “n” normally is “1" to “99") is used to specify the VTAM APPLID of one or more CICS regions to be analyzed. The APPLID associated with the CICS region to be analyzed should be entered in the **APPLIDn** guidance variable.

For example, if you wished to analyze a single CICS region named CICSPROD, you would specify **%LET APPLID1 = CICSPROD;** in USOURCE(CICGUIDE). CPExpert uses the APPLID to select data from a performance data base. Data from the performance data base will be rejected unless the APPLID in the data records matches the VTAM APPLID specified for the **APPLIDn** guidance variable(s).

The following provides guidance concerning the APPLIDn guidance variable:

- Unless the ALL_CICS guidance variable is specified as **%LET ALL_CICS=Y;** in USOURCE(CICGUIDE), you must uniquely identify the APPLID of every CICS region you wish CPExpert to analyze.

- The APPLIDn variable is ignored if the ALL_CICS guidance variable is specified as %LET ALL_CICS=Y; in USOURCE(CICGUIDE).

- The APPLIDn guidance variables need not be numerically sequential, nor must every number be specified. This provides flexibility to add or delete CICS regions from analysis without concern that the APPLIDn guidance are in any particular order. WARNING: CPExpert does not check for duplicate APPLIDn variables. The last specified APPLIDn is the operative one.

- You can temporarily "disable" the analysis of one or more CICS regions by simply replacing the "%" character before the LET with an "*" character, thus signifying to SAS that the statement is a comment.

Alternatively (and often a better choice), is to simply specify **%LET ALL_CICS=Y;** in USOURCE(CICGUIDE) and CPExpert will analyze every region that it encounters. This option does, however, add additional processing time and would not be appropriate if you wanted to analyze only a few important regions.

Chapter 1.14: MAXAPPL variable

As mentioned in the above discussion of the APPLIDn variable, the default maximum values for “n” range between 1-99 (that is, APPLID1 to APPLID99 can be specified). The MAXAPPL variable is used to limit the number of CICS regions that will be analyzed by the CICS Component during a single execution. A limit on the number of CICS regions is desired because additional code gets generated by CPExpert for each additional CICS region that is analyzed, and the number of regions to be analyzed acts a control variable in various loops within SAS coding.

Some users of CPExpert wish to analyze more than 99 CICS regions in a single execution of the CICS Component, and they wish to specify unique guidance for a subset of the CICS regions. This increased analysis can be accomplished by changing the MAXAPPL

variable to specify the maximum number of CICS regions to be analyzed. For example, to specify that a maximum of 200 CICS regions should be analyzed, specify:

```
%LET MAXAPPL = 200 ; * ANALYZE A MAXIMUM OF 200 CICS REGIONS;
```

WARNING: More than 99 regions can be analyzed concurrently only if you are executing under SAS Release 8 or above. Certain variables (particularly array element names) will be generated with a variable name larger than eight characters. SAS Release 6 does not support variable names longer than eight characters.

Chapter 1.15: LISTOFF and LISTGDE variables

CPExpert normally provides "administrative" listings that describe data and analysis parameters, before the CICS Component produces the results from its analysis.

- CPExpert normally lists any rules that have been "turned off" so uses can appreciate why the related analysis has not been done. CPExpert can "turn off" rules because there is insufficient data to execute the rules, because the rules do not apply to the release of CICS being analyzed, etc. Additionally, users can "turn off' rules for a variety of reasons.

- CPExpert normally lists the guidance contained in USOURCE(CICGUIDE). This is done because some users dynamically alter guidance based on the system or VTAM APPLID being analyzed.

Some sites use the CICS Component to analyze a very large number of CICS regions, and the analysis is performed each day. A very large amount of output can be generated because of the large number of CICS regions that are analyzed.

There generally is no change with respect to the rules that are "turned off" for any region, and there generally is no change to the guidance provided to the CICS Component. Analysts at these sites do not wish to see a daily listing of the rules that are "turned off" since the same rules generally are suppressed, and they do not wish to see a listing of the values contained in USOURCE(CICGUIDE).

The **LISTOFF** and **LISTGDE** variables allow these sites to significantly reduce the number of lines of output produced by the CICS Component.

- Specifying **%LET LISTOFF=N;** causes CPExpert to suppress the listing of the rules that have been "turned off" for a particular CICS region.

- Specifying **%LET LISTGDE=N;** causes CPExpert to suppress the listing of the guidance contained in USOURCE(CICGUIDE).

Chapter 1.16: Produce summary of rules - CICSUMRY variable

The optional CICSUMRY variable is used to produce a summary of the rules that apply to all CICS regions.

Some CPExpert users have a large number of CICS regions that are analyzed by the CICS Component. These users wish to analyze each CICS region individually, but also wish to have a summary produced that shows each rule and lists the CICS regions to which the rule applies. The CICSUMRY guidance variable can be used to produce such a summary of all findings.

The default for the CICSUMRY is “N”, indicating that no summary of rules is produced. You can specify **%LET CICSUMRY = Y**; in USOURCE(CICGUIDE). This specification will cause CPExpert to generate a listing of all rules that were produced and to list the CICS regions for which the rule applies.

The output when CICSUMRY=Y is specified is similar to the output when ALL_CICS is specified. Consequently, the CICSUMRY variable is ignored when ALL_CICS=Y is specified.

Chapter 1.17: Produce summary by system - CICS_SYS variable

The optional CICS_SYS variable is used to produce a summary of the rules by system.

Some CPExpert users have a large number of CICS regions operating on a large number of MVS systems. These users wish to analyze each CICS region individually, but also wish to have a summary produced by system, that shows each rule and lists the CICS regions to which the rule applies. The CICS_SYS guidance variable can be used to produce such a summary of all findings.

The default for the CICS_SYS is “N”, indicating that no summary of rules is produced. You can specify **%LET CICS_SYS = Y**; in USOURCE(CICGUIDE). This specification will cause CPExpert to generate a listing of all rules that were produced, summarized by MVS system, and to list the CICS regions for which the rule applies.

Chapter 1.18: SAS Output Delivery System

Output from CPExpert is created using Basic SAS statements. This Basic SAS output is designed for a standard SAS printer (line) format. With SAS Release 8, SAS users can

use the SAS Output Delivery System to create output that is formatted in Hypertext Markup Language (HTML). This output can be browsed with Internet Explorer, Netscape, or any other browser that fully supports the HTML 3.2 tag set.

Please reference the *CPExpert Installation Guide* for more detailed information about using the SAS ODS feature of CPExpert.

Some additional options have been implemented for the CICS Component. These options allow users to have HTML output produced on a system-by-system basis.

Chapter 1.18.1: SAS Output Delivery System - SYS_FRAM variable

The optional SYS_FRAM variable is applicable only if (1) you exercise the SAS ODS features, and (2) you wish to have HTML output produced on a system-by-system basis.

Some CPExpert users have a large number of CICS regions operating on a large number of MVS systems. These users wish to analyze each CICS region individually, but also wish to have a summary produced by system. The summary shows each rule produced by system, and lists the CICS regions to which the rule applies. As described earlier, the CICS_SYS guidance variable can be used to produce such a summary of all findings.

When the SAS ODS feature is used, the "summary by system" information can be appended to the files (PDSE members) identified by the **FRAME, CONTENTS,** and **BODY** guidance variables (as described above). Alternatively, the "summary by system" information can be placed in its **own file** if users producing a "summary by system" wish the summary to be directed to a different location than the normal output produced by the CICS Component.

The SYS_FRAM guidance variable serves two purposes when the SAS ODS feature is used and if CICS_SYS=Y has been specified:

- If the SYS_FRAM variable is blank (null), the frame, contents, and body information related to the "summary by system" is **appended to** the files specified by the **FRAME**, **CONTENTS**, and **BODY** variables, respectively. The FRAME, CONTENTS, and BODY variables were described previously.

- If the SYS_FRAM guidance variable is not blank, the frame, contents, and body information related to the "summary by system" is **written to** the files specified by the **SYS_FRAM**, **SYS_CONT**, and **SYS_BODY** variables[2], respectively.

  In this case, the SYS_FRAM variable identifies the file (PDSE member) that integrates the HTML table of contents and the HTML body file for the "summary by system" report.

[2] The SYS_CONT and SYS_BODY variables are described below.

A table of contents and the body file are displayed when the frame file is opened using a standard web browser.

However, if the SYS_FRAM variable is identical to the FRAME variable and the SYS_PATH variable (described below) is null or is identical to the PATH variable, the frame, contents, and body information related to the "summary by system" is **appended to** the files specified by the **FRAME**, **CONTENTS**, and **BODY** variables, respectively. This prevents "over writing" the contents of the FRAME, CONTENTS, and BODY information with the "summary by system" information.

The default value used by the CICS Component for the SYS_FRAM variable is **null**, indicating that the "summary by system" information should be appended to the file (PDSE member) described by the FRAME, CONTENTS, and BODY variables. You can alter this default by changing the value of the SYS_FRAM variable. This name is where you would point your browser to examine output created by the CICS Component.

Chapter 1.18.2: SAS Output Delivery System - SYS_PATH variable

The SYS_PATH variable is applicable only if you exercise the SAS ODS features, and if you wish to have HTML output produced on a "summary by system basis.

As described earlier, the PATH variable specifies the file name (DDNAME) of the external file where all HTML output produced by the CICS Component normally would be placed. This file specification would be the same as the DDNAME name used when creating the partitioned data set as described above.

You might wish to place the "summary by system" information in a different location than that specified by the PATH variable. In this case, you can use the SYS_PATH variable to specify a different external file for the "summary by system" information.

If the SYS_FRAM variable is non-blank but the SYS_PATH variable is null, CPExpert will use the file name specified for the PATH variable to place the "summary by system" information.

If the SYS_FRAM variable is blank but the SYS_PATH variable is not equal to the PATH variable, CPExpert will use the file name specified by the SYS_PATH variable, and place SYS_FRAM, SYS_CONT, and SYS_BODY HTML information in the location identified by the SYS_PATH variable.

Chapter 1.18.3: SAS Output Delivery System - SYS_CONT variable

The SYS_CONT variable is applicable only if you exercise the SAS ODS features, and if you wish to have HTML output produced on a "summary by system basis.

The SYS_CONT variable identifies the file (PDSE member) that contains a table of contents to the HTML output from the CICS Component for the “summary by system” information. The contents file links to the body file, which holds the actual “summary by system” output from the CICS Component.

If the SYS_CONT variable is blank (null), CPExpert will use the value specified by the CONTENTS variable.

Chapter 1.18.4: SAS Output Delivery System - SYS_BODY variable

The SYS_BODY variable is applicable only if you exercise the SAS ODS features, and if you wish to have HTML output produced on a system-by-system basis.

The SYS_BODY variable identifies the file (PDSE member) that contains the HTML output from the CICS Component.

If the SYS_BODY variable is blank (null), CPExpert will use the value specified by the BODY variable.

Chapter 2: Analysis Guidance Variables

The analysis guidance variables allow you to provide guidance to the CICS Component as CPExpert applies the CICS analysis rules. The CPEXPERT.USOURCE(CICGUIDE) module contains defaults for each guidance variable. These defaults may be appropriate for the analysis performed by the CICS Component. However, you may have unique situations (or you may simply disagree with the defaults selected).

This chapter describes the analysis guidance variables and their defaults. **Do not hesitate to make changes if the defaults for the analysis guidance variables do not meet your needs.**

Please contact Computer Management Sciences if the guidance variables are inadequate for your needs.

Exhibit 3-2 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the analysis guidance variables.


```
*********************************************************************;
*                  ANALYSIS GUIDANCE VARIABLES                      ;
*********************************************************************;
%LET PRODTEST = PROD    ; * PRODUCTION OR TEST REGION                ;
%LET GUIDE    = N       ; * OPTION TO PROVIDE INDIVIDUAL GUIDANCE    ;
%LET applid_x = GUIDE   ; * OVERRIDE GUIDANCE EXISTS FOR applid_x    ;
%LET RPTCLASS = N  ;    * CICS REPORT CLASSES ARE TO  BE ANALYZED   ;
%LET RPRTn    = report_class_name; * REPORT CLASS NAME TO ANALYZE    ;
%LET RPTGUIDE = N       ; * OPTION TO PROVIDE INDIVIDUAL GUIDANCE    ;
%LET rpt_class= GUIDE   ; * OVERRIDE GUIDANCE EXISTS FOR rpt_class_x ;
%LET CICnnn   = OFF     ; * EXAMPLE: TURN OFF CICnnn RULE            ;
%LET MAXTASK  =1        ; * TIMES MXT VALUE REACHED                  ;
%LET AMAXTASK =Y        ; * TEST WHETHER AMXT VALUE REACHED           ;
%LET CMAX1    =1        ; * TIMES CMXT VALUE REACHED - CLASS 01       ;
%LET CMAX2    =1        ; * TIMES CMXT VALUE REACHED - CLASS 02       ;
%LET CMAX3    =1        ; * TIMES CMXT VALUE REACHED - CLASS 03       ;
%LET CMAX4    =1        ; * TIMES CMXT VALUE REACHED - CLASS 04       ;
%LET CMAX5    =1        ; * TIMES CMXT VALUE REACHED - CLASS 05       ;
%LET CMAX6    =1        ; * TIMES CMXT VALUE REACHED - CLASS 06       ;
%LET CMAX7    =1        ; * TIMES CMXT VALUE REACHED - CLASS 07       ;
%LET CMAX8    =1        ; * TIMES CMXT VALUE REACHED - CLASS 08       ;
%LET CMAX9    =1        ; * TIMES CMXT VALUE REACHED - CLASS 09       ;
%LET CMAX10   =1        ; * TIMES CMXT VALUE REACHED - CLASS 10       ;

/* START TCLASS MAXACTIVE GUIDANCE FOR CICS VERSION 4.1 AND CICS/TS
 tclassname=guidance
*/

%LET ALLOCF   =0        ;  * ACCEPTABLE FAILED ALLOCATE REQUESTS      ;
%LET ALLOCQ   =1        ;  * ACCEPTABLE QUEUED ALLOCATE REQUESTS      ;
%LET CPUWARN = 40%      ;  * PCT CPU USE FOR WARNING MESSAGE (CIC700) ;
%LET DLIWAIT  =0        ;  * NO. TASKS WAITING FOR DL/I THREADS       ;
%LET ENQPOOL  =75       ;  * PERCENT ENQ CONTROL BLOCK POOL USED      ;
%LET FCGETUPD =500      ;  * MINIMUM FILE CONTROL GET UPDATE THRESHOLD ;
%LET GETMAIN  =25       ;  * NUMBER OF GETMAINS PER TASK              ;
```

SAMPLE DISPLAY OF CPEXPERT.USOURCE(CICGUIDE) MODULE

EXHIBIT 3-2


```
%LET JCBUFUL  =0        ;  * USER JOURNAL 'BUFFER FULL' CONDITIONS     ;
%LET LSRHITD  =40       ;  * LOOK-ASIDE HIT RATIO, DATA BUFFERS        ;
%LET LSRHITI  =80       ;  * LOOK-ASIDE HIT RATIO, INDEX BUFFERS       ;
%LET LSRINACT =1000     ; * HITS LESS THAN THIS INDICATES INACTIVE POOL;
%LET LSRIOREQ =75       ; * MINIMUM PERCENT I/O REQUESTS LSR FILES     ;
%LET LSRUSE   =100      ; * HITS THAT INDICATES POOL IS SELDOM USED    ;
%LET LSRUSE1  =&LSRUSE  ; * HITS THAT INDICATES POOL 1 IS SELDOM USED  ;
%LET LSRUSE2  =&LSRUSE  ; * HITS THAT INDICATES POOL 2 IS SELDOM USED  ;
%LET LSRUSE3  =&LSRUSE  ; * HITS THAT INDICATES POOL 3 IS SELDOM USED  ;
%LET LSRUSE4  =&LSRUSE  ; * HITS THAT INDICATES POOL 4 IS SELDOM USED  ;
%LET LSRUSE5  =&LSRUSE  ; * HITS THAT INDICATES POOL 5 IS SELDOM USED  ;
%LET LSRUSE6  =&LSRUSE  ; * HITS THAT INDICATES POOL 6 IS SELDOM USED  ;
%LET LSRUSE7  =&LSRUSE  ; * HITS THAT INDICATES POOL 7 IS SELDOM USED  ;
%LET LSRUSE8  =&LSRUSE  ; * HITS THAT INDICATES POOL 8 IS SELDOM USED  ;
%LET NONUSRBF =0        ; * NON-USER INITIATED BUFFER WRITES           ;
%LET NONUSRBF =0        ; * NON-USER INITIATED BUFFER WRITES           ;
%LET PCTD2TCB =80       ; * PCT NUMBER OF TCBS APPROACHING TCB LIMIT   ;
%LET PCTD2THR =80       ; * PCT NUMBER OF THREADS APPROACHING LIMIT    ;
%LET PCTEDSA  =75%      ; * PCT EDSA USED BY PEAK EDSA USAGE           ;
%LET PCTMXTHI =100%     ; * PCT MXT (ACTIVE TASKS FOR CIC108)          ;
%LET PCTMXTLO =75%      ; * PCT MXT (ACTIVE TASKS FOR CIC104)          ;
%LET PCTQRTCB =60%      ; * PCT QR TCB CPU TIME FOR RULE CIC109        ;
%LET PSBWAIT  =0        ; * NUMBER OF WAITS FOR PSB POOL BUFFERS       ;
%LET PVCOUNT  =0        ; * ACCEPTABLE NUMBER OF PV TIMEOUTS           ;
%LET RAPOOL   =0        ; * VTAM REACHED MAX RAPOOL VALUE              ;
%LET SNTCOUNT =0        ; * ACCEPTABLE NUMBER OF SNT TIMEOUTS          ;
%LET STORDUMP =1        ; * STORAGE DUMP GUIDANCE VARIABLE             ;
%LET STRWAIT  =5        ; * PERCENT I/O WAITING FOR VSAM STRINGS       ;
%LET TRANSERR =0        ; * TRANSACTION ERROR GUIDANCE VARIABLE        ;
%LET TRMSKDEL =50       ; * ACCEPTABLE SHIPPED TERMINAL DELETES        ;
%LET TSIOWAIT =5        ; * TEMP STORAGE I/O WAITING FOR BUFFERS       ;
```

SAMPLE DISPLAY OF CPEXPERT.USOURCE(CICGUIDE) MODULE

EXHIBIT 3-2 (Continued)

Chapter 2.1: Specifying Production or Test Region

The **PRODTEST** variable tells the CICS Component whether the CICS region is a production region or is a test region. The analysis performed by the CICS Component varies, depending upon whether the CICS region is a production or test. Some situations may be acceptable in a CICS test region, but would be unacceptable in a CICS production region.

For example, storage violations may be acceptable (and even expected) in a CICS test region. However, a storage violation in a CICS production region should be cause for immediate action. The CICS Component would essentially ignore any storage violations if the CICS region being analyzed were a test region. However, any storage violation detected in a production CICS region would generate a strong recommendation.

Specify **%LET PRODTEST=TEST;** to change the guidance from the default CICS production region classification to a CICS test region classification.

You may wish CPExpert to analyze both production and test regions during a single execution. In this case, you should specify **%LET PRODTEST=PROD** (the default), to ensure that the production regions are properly analyzed. Some of the results from analyzing the CICS test regions may not be relevant (since CPExpert will be analyzing the test regions as though they were production regions, and thus apply more stringent criteria). You can simply discount these results.

Chapter 2.2: Specifying guidance for regions - GUIDE variable

The **GUIDE** guidance variable specifies whether override guidance exists for one or more CICS regions. Please see Section 3 (Chapter 3) for a detailed description of how to specify override guidance for one or more CICS regions.

Chapter 2.3: Identifying regions with guidance - applid_x=GUIDE

The **applid_x=GUIDE** guidance identifies specific regions that have unique guidance. Please see Section 3 (Chapter 3) for a detailed description of how to specify override guidance for one or more CICS regions.

Chapter 2.4: Analyze CICS report classes - RPTCLASS variable

The **RPTCLASS** guidance variable acts as a “switch” to specify whether the CICS Component should analyze CICS Report Classes, based on information contained in RMF Type70(series) records.

Some users of the CICS Component want CPExpert to provide a "warning" about certain system-level conditions related to CICS workload. To meet this requirement, the CICS Component has been enhanced to provide an ability to associate CICS workload with system-level information contained in RMF Type70(series) records.

The initial requirement was for CPExpert to provide a warning when certain CICS workload used more than a specified percent of CPU. For this requirement, logic was implemented to associate CICS workload with the TYPE72GO information. Guidance was implemented using the CPUWARN guidance variable. While other features may be implemented, the "CPU warning" construct will be used to discuss and illustrate both the concept and the implementation of the enhancement.

The "CPU warning" approach requires that a Report Class describing the CICS workload of interest be defined to the Workload Manger (WLM), using the standard WLM workload classification panel. The System Resources Manager (SRM) acquires information related to the defined Report Class, and RMF writes the Report Class information to SMF as part of the SMF Type 72 records. MXG processes the SMF Type 72 records and places information into various files in the MXG performance data base. CPExpert processes the MXG TYPE72GO file to extract information about the defined Report Class. CPExpert produces a "CPU warning" when the defined Report Class uses more than nn% CPU during any RMF reporting interval.

A CPExpert user implements the "CPU warning" feature by the following steps:

- **Define Report Class**. The "Defining Report Classes" section of IBM's *MVS Planning: Workload Management* document describes how to define report classes to the WLM. In brief, the WLM's classification rules can assign incoming work to a report class. Report classes can be used to report on a subset of transactions running in a single service class, or can report on a combination of transactions running in different service classes within one report class.

- **Specify that report classes are to be analyzed by the CICS Component**. The **RPTCLASS** guidance variable is used to tell CPExpert that report classes are to be analyzed. If you wish the CICS Component to analyze report classes, specify **%LET RPTCLASS=Y;** in USOURCE(CICGUIDE).

- **Identify the report classes to be analyzed**. You must tell CPExpert which report classes are to be analyzed. This is done by specifying %LET RPTn=report_class_name; in USOURCE(CICGUIDE). For example, to tell CPExpert that the CICSAOR1 report class should be analyzed, specify **%LET RPT1=CICSAOR1;** in USOURCE(CICGUIDE). You can specify up to 99 report classes to be analyzed.

Once the above process has been implemented, the CICS Component will extract appropriate information from the MXG TYPE72GO file, compare the information with the

guidance provided, and produce rules in the CIC700(series) if the guidance has been exceeded.

Chapter 2.5: CICS report classes to analyze - RPRTn variable

The **RPRTn** (where “n” normally is “1" to “99") guidance identifies CICS Workload Manager report classes that should be analyzed by the CICS Component. The report class name associated with the CICS workload to be analyzed should be entered in the **RPRTn** guidance variable. Please see Section 3 (Chapter 4) for a detailed description of how to analyze CICS workload in one or more CICS report classes.

For example, if you wished to analyze a single CICS report class named CICSAOR3, you would specify **%LET RPRT1 = CICSAOR3;** in USOURCE(CICGUIDE). In this example, CPExpert uses the RPRT1 guidance variable to select information for the CICSAOR Report Class from the MXG TYPE72GO file.

The following provides guidance concerning the RPRTn guidance variable:

- You must uniquely identify the report class name of every CICS report class you wish CPExpert to analyze.

- The RPRTn guidance variables need not be numerically sequential, nor must every number be specified. This provides flexibility to add or delete CICS report classes from analysis without concern that the RPRTn guidance variables are in any particular order.

- You can temporarily "disable" the analysis of one or more CICS report classes by simply replacing the "%" character before the LET with an "*" character, thus signifying to SAS that the statement is a comment.

Chapter 2.6: Specifying guidance for CICS report classes - RPTGUIDE variable

The **RPTGUIDE** guidance variable specifies whether override guidance exists for one or more CICS report classes. Please see Section 3 (Chapter 4) for a detailed description of how to specify override guidance for one or more CICS report classes.

Chapter 2.7: Identifying CICS report classes with guidance - report_class_x=GUIDE

The **report_class_x=GUIDE** guidance identifies specific CICS report classes that have unique guidance. Please see Section 3 (Chapter 4) for a detailed description of how to specify override guidance for one or more CICS report classes.

Chapter 2.8: Turning OFF CICS Component Rules

The default guidance values for the CICS Component are specified based on either IBM's guidance contained in IBM's *CICS Performance Guides*, or based on guidance from industry sources. For most findings, users can use analysis guidance variables to control the analysis; rule results will be produced only when situations exceed the guidance provided.

However, some users of the CICS Component wish to suppress the analysis and findings of particular rules, or suppress the analysis and findings only for particular CICS regions. This desire typically is caused by (1) an overall disagreement with the finding, (2) an inability to make a suggested change, or (3) a decision that a particular finding is inapplicable to a particular CICS region.

Regardless of the reason for wishing to suppress particular findings by the CICS Component, users wish the ability to "turn off" certain rules.

All rules are ON by default, although the CICS Component may turn rules OFF if insufficient data exists to perform analysis or if the rule does not apply to the release of CICS being analyzed.

Rules can be turned OFF by specifying **%LET CICnnn = OFF;** , where "nnn" is the rule number that you wish to turn OFF. This specification should be placed in the *ANALYSIS GUIDANCE SECTION* of USOURCE(CICGUIDE).

For example, you can turn OFF Rule CIC104 by specifying **%LET CIC104=OFF;** in the *ANALYSIS GUIDANCE SECTION* of USOURCE(CICGUIDE).

If you are analyzing multiple CICS APPLIDs during a single execution of the CICS Component and you are specifying specific guidance for particular CICS regions, you can turn rules OFF (or turn rules ON) for particular regions. For example, you may wish to turn OFF a rule normally, but wish to have that rule ON for a particular CICS region. Simply specify **%LET CICnnn=OFF;** in USOURCE(CICGUIDE), where "nnn" is the rule you wish to turn OFF. Then you can specify **%LET CICnnn=ON;** in the USOURCE(applid_x) member applying to the particular CICS region for which the rule should be ON.

Please note that the CICS Component verifies that all required data is present in your performance data base before invoking each rule. A rule will be suppressed if any required data is missing, regardless of your specification to suppress or enable the rule.

Chapter 2.9: Times at MXT - MAXTASK variable

The MXT operand in the System Initialization Table (SIT) limits the total number of concurrent tasks in the CICS region. Beginning with CICS Release 1.7, CICS will not create a Task Control Area (TCA) for a new task when the number of tasks concurrently in the region reaches the MXT value. The MXT value (from the SIT or as modified with operator overrides) is contained in variable XMGMXT in the MXG file CICXMG.

CPExpert produces Rule CIC101 if the CICS interval statistics reported that the maximum tasks value (contained in variable XMGMXT) was reached more than the **MAXTASK** guidance variable and CPExpert did not detect a storage constraint.

The default specification for the guidance variable is **%LET MAXTASK = 1**. This value indicates that CPExpert will produce Rule CIC101 if the CICS region reached MXT more than once. You can provide different guidance to CPExpert by changing the MAXTASK if you feel that Rule CIC101 is produced spuriously.

Chapter 2.10: Times at AMXT - AMAXTASK variable

The AMXT operand in the System Initialization Table (SIT) limits the total number of concurrent **active** tasks in the CICS region. All new and resumed tasks must pass the AMXT limit before CICS dispatches the tasks. CICS marks all new and resumed tasks "nondispatchable for AMXT reasons." The AMXT value (from the SIT or as modified with operator overrides) is contained in variable DSGAMXT in the MXG file CICDS.

CPExpert produces Rule CIC102 if the CICS statistics reported that the AMXT value (contained in variable DSGAMXT) was reached and CPExpert did not detect a storage constraint.

The default specification for the guidance variable is **%LET AMAXTASK = Y**. This specification indicates that CPExpert will produce Rule CIC102 if the number of active tasks in the CICS region was as high as the AMXT value. You can "turn off" this rule by specifying **%LET AMAXTASK = N;** if you feel that Rule CIC102 is produced prematurely.

Chapter 2.11: Times at CMXT - CMAXn variables

This guidance variable is applicable only for CICS Releases prior to CICS/ESA 4.1 (beginning with CICS Release CICS/ESA 4.1, the MAXACTIVE attribute for a transaction class is used).

The Maximum Task Class (CMXT) operand in the System Initialization Table (SIT) limits the total number of concurrent **active** tasks associated with particular task classes. Up to 10 unique task classes can be defined, and a maximum number of active tasks is specified for each class. The CMXT value (from the SIT or as modified with operator overrides) for each task class is contained in variable A15MXTM in the MXG file CICTCLR.

CPExpert provides guidance variables for each task class. These guidance variables are CMAX1 through CMAX10, corresponding to Task Class 1 through Task Class 10, respectively.

CPExpert produces Rule CIC103 if the CICS statistics reported that the maximum active tasks value (contained in variable A15MXTM) for any class was reached more than the CMXT guidance value for the class, and if the maximum number of concurrently attached tasks for the task class was greater than the CMXT specification for the task class.

The default specification for the guidance variables is **%LET CMAXn = 1**, where "n" is the appropriate class task. This value indicates that CPExpert will produce Rule CIC103 if the CICS region reached CMXT limit more than once for the particular class, and if additional tasks were attached after the limit was reached. You can provide different guidance to CPExpert by changing the CMAXn variables if you feel that Rule CIC103 is produced prematurely.

Chapter 2.12: Times at MAXACTIVE - *tclassname* variables

This guidance variable applies with CICS/ESA Release 4.1, CICS Transaction Server for OS/390, or CICS Transaction Server for z/OS.

The MAXACTIVE attribute for a transaction class can be used to control the number of active tasks in the transaction class. The MAXACTIVE value (from the SIT or as modified with operator overrides) for each transaction class is contained in variable XMCMXT in the MXG file CICXMC.

The MAXACTIVE attribute limits the number of transactions for a specific transaction class, while the MXT value (specified in the System Initialization Table) limits the total number of transactions in the CICS region. Please refer to Rule CIC105 for additional discussion of the MAXACTIVE attribute.

Prior to CICS/ESA Release 4.1, the CMXT values were used to limit the number of active tasks in a transaction class, and up to 10 transaction tasks could be defined (numbered 1-10). With CICS/ESA Release 4.1, a transaction class is given a **name**, rather than a **number**. Consequently, the guidance to CPExpert is different with CICS/ESA Release 4.1.

CPExpert produces Rule CIC105 if the CICS interval statistics reported that the MAXACTIVE value (contained in variable XMCMXT) for any class was reached more than the guidance value for the class. The default for the guidance value is "1", indicating that Rule CIC105 will be produced if the MAXACTIVE value was reached more than one time. This default may cause Rule CIC105 to be produced for a transaction class when you have explicitly constrained the maximum number of tasks in a transaction class. Consequently, you may wish to tell CPExpert not to produce Rule CIC105 unless the MAXACTIVE value is exceeded a much larger number of times.

Initially, CPExpert provided guidance variables for each transaction class, using the **transaction class name** as the guidance variable. For example, suppose that "CLASSA" had been specified as the name for a transaction class. The guidance variable to CPExpert would have been specified as **%LET CLASSA = n** where "n" is the guidance for CLASSA. This process created SAS macro variables as the TCLASS name, and the value of the macro variable was the guidance. CPExpert would produce Rule CIC105 if the MAXACTIVE value was reached more than "n" times for the transaction class. Sadly, this approach did not always work.

CICS transaction class names can have special characters “@, #, and $” in the TCLASS name. Unfortunately, SAS will not accept these special characters imbedded in a SAS macro variable name. Consequently, a different approach was designed.

Exhibit 3-3 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the analysis guidance variables. As illustrated in Exhibit 3-3, guidance for the TCLASS MAXACTIVE is specified **inside** the SAS macro comment statements (/* and */). The specification is “tclassname = guidance” where “tclassname” is the name of the transaction class and the “guidance” is the number of times that the transaction is allowed to reach MAXTASK before Rule CIC105 is produced. For example:

```
/* START TCLASS MAXACTIVE GUIDANCE
   $067 = 50
*/
```

would be used to specify that CPExpert should produce Rule CIC105 when the $067 transaction class reached MAXACTIVE more than 50 times during a CICS interval statistics period.

Chapter 2.13: Failed ALLOCATE Requests - ALLOCF variable

A CICS region can communicate with another CICS region or with a remote system only after a "communication link" has been established between the CICS region and the other CICS region or the remote system. A part of the communication link is the allocation of "sessions" which can send or receive requests. Multiple sessions can be active on the communication link, and the sessions are defined to CICS as "send" sessions or "receive" sessions. (Please refer to Rule CIC260 for more discussion of communication links and sessions.)

Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command. With LU6.2, ALLOCATE requests can be for **any** modegroup defined with the connection (a **generic** allocation request), or may be for a **particular** modegroup defined with a connection (a **specific** allocation request).

CICS will process a session ALLOCATE request only if the communication link is established and operational between the two systems. CICS will reject the ALLOCATE request if the connection has been released, the connection is out of service, or the mode group has been closed.

CPExpert produces Rule CIC265 if the number of Failed Link Allocates (A14ESTAF) is greater than the **ALLOCF** guidance variable, for requests to allocate **generic** sessions. CPExpert produces Rule CIC266 if the number of Failed Link Allocates (A20ESTAF) is greater than the **ALLOCF** guidance variable, for requests to allocate **specific** sessions with a modegroup.

The default specification for the ALLOCF guidance variable is **%LET ALLOCF = 0**, indicating that no failed allocation requests are acceptable. You can provide different guidance to CPExpert by changing the ALLOCF guidance variable if you feel that Rule CIC265 or Rule CIC266 are produced prematurely.

Chapter 2.14: Unsuccessful ALLOCATEs - ALLOCQ variable

A CICS region can communicate with another CICS region or with a remote system only after a "communication link" has been established between the CICS region and the other CICS region or the remote system. A part of the communication link is the allocation of "sessions" which can send or receive requests. Multiple sessions can be active on the communication link, and the sessions are defined to CICS as "send" sessions or "receive" sessions. (Please refer to Rule CIC260 for more discussion of communication links and sessions.)

Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command. With LU6.2, ALLOCATE requests can be for **any** modegroup defined with the connection (a **generic** allocation request), or may be for a **particular** modegroup defined with a connection (a **specific** allocation request).

A session must be available in order to be allocated in response to the ALLOCATE command. If a session is not available, CICS will normally queue the allocate request (and suspend the transaction) until a session is made available or until the transaction "times out" if the DTIMOUT parameter is used in DFHPCT. Optionally (using the NOQUEUE specification), control can be returned to the transaction which can take application-dependent action based on the unavailability of a session.

Some queuing for allocation requests may be unavoidable because an installation may have deliberately restricted the number of sessions to minimize resource use by CICS. Similarly, some return of session ALLOCATE requests to the transaction because sessions were unavailable (if the NOQUEUE option were selected) may be unavoidable for the same reasons.

However, queuing of session ALLOCATE requests delays transactions and causes resources held by the transaction to be unavailable for other work. If the NOQUEUE option were selected, the transaction may implement alternative processing or simply delay the ALLOCATE request for a time interval in hopes that a session will become available. The NOQUEUE option may have less impact, depending upon the specifics of the application. In either case, it is not normally desirable to have ALLOCATE requests unsatisfied because sessions are unavailable.

CPExpert detects a **potential** problem with unavailable sessions in a variety of situations. These situations are described by rules in the CIC260(series). Most of the final decisions made by rules in the CIC260(series) compare values against the **ALLOCQ** guidance variable. The default specification for this variable is **%LET ALLOCQ = 1**, indicating that the CIC260(series) rules would be produced if more than one ALLOCATE request could not be satisfied because no sessions were available. **This low default value is intended only to alert you to a potential problem with the number of sessions defined, and is intended to make you aware of this analysis mechanism**.

For many installations, the default should be changed after executing the CICS Component a few times. The **ALLOCQ** guidance variable should normally be used to cause CPExpert to signal a problem only when you wish to be informed of abnormal situations. For example, some installations always have a few ALLOCATE requests queued. Occasionally, however, several hundred or even several thousand requests are queued. Analysts at these installations are not concerned about the few queued requests,

but are concerned about the situations when hundreds or thousands of requests are queued.

Chapter 2.15: CICS workload used excess CPU - CPUWARN variable

Some users of the CICS Component want CPExpert to provide a “warning” about certain system-level conditions related to CICS workload. To meet this requirement, the CICS Component has been enhanced to provide an ability to associate CICS workload (as identified by Workload Manager report classes) with system-level information contained in RMF Type 72 records.

The initial requirement was for CPExpert to provide a warning when certain CICS workload used more than a specified percent of CPU. For this requirement, logic was implemented to associate CICS workload with the TYPE72GO information. Guidance was implemented using the **CPUWARN** guidance variable.

CPExpert produces Rule CIC700 if the CICS workload identified by a report class used more than the percent CPU identified by the **CPUWARN** guidance variable. The default specification for the CPUWARN guidance variable is **%LET CPUWARN = 40%**. You can provide different guidance to CPExpert by changing the CPUWARN variable.

Chapter 2.16: Tasks waiting for DL/I threads - DLIWAIT variable

The DLTHRED operand in the System Initialization Table (SIT) specifies the number of concurrent DL/I threads that can be allocated for IMS/VS data bases. The number of concurrent DL/I threads limits the number of tasks concurrently scheduled for use of IMS/VS resources.

A significant amount of real and virtual storage can be required if a large number of DL/I threads are specified. However, if the number of DL/I threads is too low, tasks can wait for threads. If tasks wait for DL/I threads, they tie up storage while they are waiting, and response suffers. Therefore, there is a tradeoff between the amount of storage allocated for DL/I threads and the delays caused by tasks waiting for threads.

The CICS interval statistics report the number of times tasks waited for DL/I Threads. CPExpert produces Rule CIC190 if storage was not a constraint, and if the number of times tasks waited for DL/I Threads was greater than the **DLIWAIT** guidance value. The default specification for this guidance variable is **%LET DLIWAIT = 0**. You can provide different guidance to CPExpert by changing the DLIWAIT variable.

Chapter 2.17: IMS ENQ pool space used - ENQPOOL variable

The ENQPL operand in the SIT specifies the number of blocks in the IMS enqueue control block pool. The enqueue control block pool is used only if program-isolation scheduling is being used. If the ENQPL value is too small, the IMS/VS DB task abends with a U0775 pseudoabend message, causing dynamic backout of the changes.

Obviously, if the IMS/VS DB task abends, action would be taken to increase the ENQPL value. However, CPExpert analyzes CICS statistics to give an advance warning that such abends may occur with the present setting of the ENQPL operand.

CPExpert produces Rule CIC193 if the percent of the ENQ control block pool used was greater than the **ENQPOOL** guidance variable. The default specification for this guidance variable is **%LET ENQPOOL = 75**, indicating that Rule CIC193 will be produced when more than 75% of the ENQ control block pool is used. You can provide different guidance to CPExpert by changing the ENQPOOL variable if you feel that Rule CIC193 is produced prematurely.

Chapter 2.18: Percent unnecessary UPDATE - FCPCTUPD variable

A CICS application accesses VSAM data sets using CICS file control commands. The file control commands can optionally specify that an update is to occur with the record being read (keyword UPDATE). The UPDATE option guarantees read integrity for the record. One result of this UPDATE option is that the record is locked (and, depending on the type of file, the control interval is locked). Another result of this UPDATE option is that, if the VSAM data set is assigned to a CICS-maintained data table, the VSAM source data set must be referenced by VSAM before the record is referenced in the data table reference.

Both results from using the UPDATE option cause overhead and potentially degrade performance. Consequently, the UPDATE option should be used only if the record is actually updated or it is deleted.

File control statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed a VSAM data set using the UPDATE option but did not subsequently update or delete the records.
CPExpert produces Rule CIC177 when the percent unnecessary UPDATE option is more than the value specified by the PCTFCUPD guidance variable in USOURCE(CICGUIDE). The default value for the PCTFCUPD is 25 indicating that CPExpert should produce Rule CIC177 whenever more than 25% of the VSAM file accesses with the UPDATE option did not result in a corresponding change to the VSAM source data set.

CPExpert normally suppresses this finding if less than 500 GET UPDATE file control commands were issued against the file. You can specify a different threshold for

suppressing Rule CIC177 by altering the **FCGETUPD** guidance variable in USOURCE(CICGUIDE).

Chapter 2.19: Number of GETMAINs - GETMAIN variable

The CICS interval statistics report the number of times CICS acquired storage for a task (CICS issued a GETMAIN). The number of GETMAINs is related to the design and use of the tasks and applications. Consequently, there are no CICS tuning controls relating to the number of GETMAINs.

GETMAINs impose system overhead for the system code necessary to execute the GETMAIN. It is desirable to minimize the GETMAINs per task to minimize this overhead, although tasks obviously must issue GETMAINs to acquire any necessary storage.

CPExpert produces Rule CIC110 if the result is greater than the **GETMAIN** guidance variable in USOURCE.CICGUIDE. The default specification for this guidance variable is **%LET GETMAIN = 25**. As a "rule of thumb" indicated in IBM documents, there should be fewer than an average of 25 GETMAINs per task. More than this value generally indicates that the task may be improperly designed or that it is experiencing performance problems of some type. You can provide different guidance to CPExpert by changing the GETMAIN variable if you feel that Rule CIC110 is produced prematurely.

Chapter 2.20: "Buffer full" for user journals - JCBUFUL variable

A user journal may be used to record file control or message activity during CICS execution. CICS can be directed to perform automatic journaling by specifying entries in either the file control table or the program control, or users can provide explicit user journaling routines.

For systems prior to CICS Release 3.2.1, a single buffer is acquired for a user journal (CICS Release 3.2.1 acquires two buffers for each journal). The size of each journal buffer is specified in the BUFSIZE operand of the DFHJCT macro.

If the buffer is full, tasks using the user journal must wait. This wait time delays the tasks and lengthens response. Additionally, there is potentially a more serious effect for the overall CICS region when tasks wait for user journal, if the journal is used by several tasks. When tasks wait for a journal output, they will become dispatchable whenever the physical output completes and their records are placed in the buffer. CICS can enter a stress condition if a number of tasks are suddenly dispatchable and require resources.

CPExpert produces Rule CIC221 if the number of times a buffer full" condition exceeded the CPExpert **JCBUFUL** guidance variable. This guidance variable is provided so that the rule will not be produced spuriously in those situations when system design decisions

require user journals to be written frequently and the BUFSUV operand is insufficient to meet the requirement.

The default specification for this variable is **%LET JCBUFUL = 0**, indicating that the buffer full condition should not be encountered for user journal logs. You can provide different guidance to CPExpert by changing the JCBUFUL variable if you feel that Rule CIC221 is produced prematurely.

Chapter 2.21: Look-aside read hit - LSRHITD and LSRHITI variables

VSAM files assigned to a Local Shared Resources (LSR) pool share common buffers (and also share strings) assigned to the LSR pool. Since the buffers are shared, significantly fewer buffers normally are required to support I/O access operations.

There is an extremely important advantage to using LSR pools for VSAM files: VSAM will use its "look-aside" logic to determine whether a required control interval (CI) is already in a buffer, before executing any physical I/O operations. If the required record is already in a buffer, VSAM will use the record in storage, rather than issuing a read to DASD. This has the effect of implementing an in-storage caching of the file, and can **significantly** reduce the number of physical I/O operations required.

The CICS interval statistics provide information about the number of times an I/O request was satisfied because VSAM found the data in a LSR buffer. CPExpert calculates the percent of "look-aside hits" versus total I/O access operations (the sum of "look-aside hits" and "buffer reads"). The resulting percent is compared with one of two guidance variables, depending upon how the subpool is used.

- If the subpool is used for data records, the percent is compared to the **LSRHITD** guidance variable.

- If the subpool is used for index records, the percent is compared to the **LSRHITI** guidance variable.

The default values for the LSRHITD and LSRHITI guidance variables are 40% and 80%, respectively. If the subpool is used for both data and index records, CPExpert tests the index **LSRHITI** guidance variable first. This is done since index records should have a higher hit ratio than data records.

CPExpert produces Rule CIC165 if the percent of "look-aside hits" is lower than the appropriate guidance variable. The default specification for the guidance variables is **%LET LSRHITD = 40** (for data records) and **%LET LSRHITI = 80** (for index records). You can provide different guidance to CPExpert by changing the LSRHITD and LSRHITI variables.

Chapter 2.22: Inactive pool - LSRINACT variable

As described in the above discussion related to LSRHITI and LSRHITD, CPExpert produces Rule CIC165 if the percent of "look-aside hits" is lower than the appropriate guidance variable. However, some LSR pools are seldom used, and Rule CIC165 should not be produced for these LSR pools. Consequently, CPExpert compares the total “hits” for each pool against the LSRINACT guidance variable. The LSRINACT guidance variable specifies the threshold below which CPExpert should ignore the “look-aside hits” because the pool is seldom used.

The default specification for this guidance variable is **%LET LSRINACT = 1000**, meaning that CPExpert will ignore the percent “look-aside hits” for any LSR pool that had less than 1000 total hits. You can provide different guidance to CPExpert by changing the LSRINACT variable.

Chapter 2.23: % I/O requests for LSR files - LSRIOREQ variable

VSAM files can be assigned as nonshared resources (NSR) files or assigned to a local shared resources (LSR) pool. The major difference between the two methods is how the VSAM strings and VSAM buffers are allocated and used.

- When CICS VSAM files are assigned as NSR files, there is no sharing of VSAM strings or VSAM buffers among files.

- When CICS VSAM files are assigned to LSR pools, the files share common strings and buffers assigned to the LSR pool.

There are some advantages to assigning files as NSR files, and there are major advantages to assigning files to LSR pools. The description of Rule CIC167 describes the advantages of each technique.

CPExpert analyzes the number of I/O requests for LSR files and computes this as a percent of all I/O requests for all VSAM files. CPExpert produces Rule CIC167 if this percent is less than the LSRIOREQ guidance variable.

The default of the LSRIOREQ guidance variable is 75, indicating that 75% of the I/O requests should be satisfied from files assigned to LSR pools. This default is set such that Rule CIC167 probably will initially be produced for many CICS regions. The point of producing Rule CIC167 is to alert you to the significant performance advantages of using LSR pools. If you are not inclined to assign more VSAM files to LSR pools, change the guidance variable and prevent the spurious production of Rule CIC167. (For example, you can "turn off" the rule completely by specifying **%LET LSRIOREQ = 0**.)

Chapter 2.24: Inactive pool - LSRUSE variable

VSAM files assigned to a LSR pool share common buffers (and also share strings) assigned to the LSR pool. Since the buffers are shared, significantly fewer buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time. Rather, file accesses will tend to be distributed across files at different times. Some files will have requirements for buffers at one time, while at another time they will not be accessed and will not require buffers. The demand for buffers therefore is the **peak collective demand** rather than the **sum** of the **peak individual** demands.

The discussion regarding benefits of using LSR pools assumes that VSAM files actually use the pools. If VSAM files do not use the LSR pools that are created, the storage dedicated to a particular pool might be better used elsewhere in CICS.

CPExpert calculates the total I/O requests made to each LSR pool that was defined. The resulting value is compared with the LSRUSE guidance variable. CPExpert produces Rule CIC175 if the total number of I/O requests to the LSR pool is lower than the LSRUSE guidance variable. The default specification for the guidance variables is **%LET LSRUSE = 100**, meaning that CPExpert will produce Rule CIC175[3] when less than 100 I/O requests are made to the LSR pool. You can provide different guidance to CPExpert by changing the LSRUSE variable.

Chapter 2.25: Inactive subpool - LSRUSEn variables

LSR pools are generated using the DEFINE LSRPOOL command of Resource Definition Online, or using the DFHFCT TYPE=SHRCTL macro. The BUFFERS operand of this macro is used to define the number and size of the buffers assigned to the pool. Buffers are normally specified with different sizes, with the different sizes corresponding to the size of the CI for data and index records of files assigned to the LSR pool. The pools with different buffer sizes are referred to as LSR subpools. There can be (and should be) different LSR subpools for data and index CI sizes.

The discussion regarding benefits of using LSR pools assumes that VSAM files actually use the subpools. If VSAM files do not use the LSR subpools that are created, the storage dedicated to buffers for a particular subpool might be better used elsewhere in CICS.

The CICS statistics provide information about the number of times an I/O request was made to a LSR pool, and identify the subpool to which the request was made. CPExpert calculates the total I/O requests made to each LSR subpool that was defined. The

[3] CPExpert will not produce Rule CIC175 unless the condition exists for more than 50% of the CICS statistics intervals being analyzed.

resulting value is compared with the LSRUSEn guidance variable (where “n” is the number of the LSR pool, ranging from 1 to 8). CPExpert produces Rule CIC176 if the total number of I/O requests to the LSR subpool is lower than the LSRUSEn guidance variable.

The default value for each LSRUSEn guidance variable is based on the global LSRUSE guidance variable. The default values for the LSRUSE guidance variable is 100, meaning that CPExpert will produce Rule CIC176 when less than 100 I/O requests are made to the LSR subpool. You can use the default value for the LSRUSE guidance variable, specify a different value for this global variable (and have the global value applied to each LSR pool), or specify a different value for one or more LSR pools.

Chapter 2.26: Non-user buffer writes - NONUSRBF variable

User initiated buffer writes are the writing of buffers due to a direct request from the user program, such as EXEC CICS WRITE of a single record. This record must be written to the file immediately.

Non-user initiated buffer writes are usually encountered only when deferred requests are involved. For example, a MASSINSERT operation could cause non-user initiated buffer writes. Non-user initiated buffer writes normally mean that the number of buffers assigned to the LSR subpool should be increased.

CPExpert produces Rule CIC169 if the CICS interval statistics revealed that the number of non-user initiated buffer writes for any LSR subpool was greater than the **NONUSRBF** guidance variable. The default specification for this variable is **%LET NONUSRBF = 0**, indicating that non-user initiated buffer writes should not be encountered. You can provide different guidance to CPExpert by changing the NONUSRBF variable if you feel that Rule CIC169 is produced prematurely.

Chapter 2.27: Percent CICS-DB2 TCBs used - PCTD2TCB variable

The TCBLIMIT parameter specifies the maximum number of Task Control Blocks (TCBs) that can be used for a CICS-DB2 connection.

CPExpert produces Rule CIC275 when the percent peak TCBs in use is more than the value specified by the **PCTD2TCB** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTD2TCB** is 80 indicating that CPExpert should produce Rule CIC275 whenever the peak TCBs in use was more than 80% of the TCB limit specified for the CICS-DB2 Connection. You can provide different guidance to CPExpert by changing the PCTD2TCB variable if you feel that Rule CIC275 is produced prematurely.

Chapter 2.28: Percent CICS-DB2 threads used - PCTD2THR variable

The THREADLIMIT parameter specifies the maximum number of pool threads that can be used for a CICS-DB2 connection.

CPExpert produces Rule CIC277 when the percent pool threads in use is more than the value specified by the **PCTD2THR** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTD2THR** is 80 indicating that CPExpert should produce Rule CIC277 whenever the pool threads in use was more than 80% of the pool thread limit specified for the CICS-DB2 Connection. You can provide different guidance to CPExpert by changing the PCTD2THR variable if you feel that Rule CIC277 is produced prematurely.

Chapter 2.29: Percent EDSA used at peak usage - PCTEDSA variable

The EDSALIM keyword in the SIT specifies, in 1 megabyte increments, the upper limit of the total amount of storage within which CICS can allocate the individual extended dynamic storage areas (EDSAs) that reside above the 16MB boundary. From the storage value that is specified on the EDSALIM parameter, CICS allocates the extended user DSA (EUDSA), the extended read-only DSA (ERDSA), the extended shared DSA (ESDSA), and the extended CICS DSA (ECDSA).

When the amount of storage allocated by CICS reaches the limit specified by the EDSALIM keyword, the result can be increased program compression or, more seriously, SOS (short on storage) conditions, or even storage deadlock ABENDS when program compression is not sufficient.

The EDSALIM value (from the SIT or as modified with operator overrides) is contained in variable SMSEDSAL in the MXG file CICSMDSA. The CICS interval statistics contain information about the amount of EDSA allocated.

CPExpert compares the peak EDSA allocated against a percent of the value specified for the EDSALIM keyword (contained in variable SMSEDSAL). CPExpert produces Rule CIC113 when the peak EDSA allocated storage is greater than this percent. The percent used by CPExpert is specified by the **PCTEDSA** guidance variable in USOURCE(CICGUIDE).

The default value for the PCTEDSA guidance variable is 75%, indicating that Rule CIC113 should be produced when the peak EDSA allocated storage is above 75% of the value of the EDSALIM keyword in the SIT. You can provide different guidance to CPExpert by changing the PCTEDSA guidance variable if you feel that Rule CIC113 is produced prematurely.

Chapter 2.30: Percent EDSA used at peak usage - PCTEDSA variable

A CICS application accesses VSAM data sets using CICS file control commands. The file control commands can optionally specify that an update is to occur with the record being read (keyword UPDATE). The UPDATE option guarantees read integrity for the record. One result of this UPDATE option is that the record is locked (and, depending on the type of file, the control interval is locked). Another result of this UPDATE option is that, if the VSAM data set is assigned to a CICS-maintained data table, the VSAM source data set must be referenced by VSAM before the record is referenced in the data table reference.

Both results from using the UPDATE option cause overhead and potentially degrade performance. Consequently, the UPDATE option should be used only if the record is actually updated or it is deleted.

File control statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed a VSAM data set using the UPDATE option but did not subsequently update or delete the records.
CPExpert produces Rule CIC177 when the percent unnecessary UPDATE option is more than the value specified by the **PCTFCUPD** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTFCUPD** is 25 indicating that CPExpert should produce Rule CIC177 whenever more than 25% of the VSAM file accesses with the UPDATE option did not result in a corresponding change to the VSAM source data set.

CPExpert normally suppresses this finding if less than 500 GET UPDATE file control commands were issued against the file. You can specify a different threshold for suppressing Rule CIC177 by altering the FCGETUPD guidance variable in USOURCE(CICGUIDE).

Chapter 2.31: Active Tasks as Percent of MXT (PCTMXTHI)

The MXT operand in the System Initialization Table (SIT) limits the total number of concurrent tasks in the CICS region. Beginning with CICS Release 1.7, CICS will not create a Task Control Area (TCA) for a new task when the number of tasks concurrently in the region reaches the MXT value.

Specifying a “correct” value for the MXT keyword is a balance between (1) specifying a value that is too high and (2) specifying a value that is too low.

- If the MXT value is too high, storage is wasted and (with Goal Mode) unnecessary overhead is generated. The logic associated with Rule CIC104 deals with situation in which the MXT value is too high.

- If the MXT value is too low, CICS will fail to attach tasks when the number of tasks reaches the MXT value. Rule CIC101 provides an indication of the number of times that the number of active tasks reached the MXT value.

  This indication that MXT was reached is acceptable for many CICS regions. However, reaching MXT might be unacceptable for some critical CICS regions. Installation personnel need to be aware of the possibility of a too-low MXT value for critical CICS regions. The logic associated with Rule CIC108 can be used to provide an alert that MXT value might be too low.

CPExpert computes the percent of active tasks as a function of the value of the MXT keyword. When the percent of active tasks is greater than the **PCTMXTHI** guidance variable, CPExpert concludes that the MXT value may be too low. Rule CIC108 is produced to report this conclusion.

The value of the PCTMXTHI guidance variable can be specified either as “nn%” or as “.nn” to indicate a percentage. The default value for the PCTMXTHI guidance variable is 100%, which effectively “turns off” the logic in Rule CIC108 (the percent cannot be higher than 100%). You can specify an appropriate percentage if you have critical CICS regions and wish to be alerted that the number of active tasks may be in danger of reaching the MXT specification. CPExpert will produce Rule CIC108 when the number of active tasks reaches the specified percent of MXT.

Chapter 2.32: Active Tasks as Percent of MXT (PCTMXTLO)

The value of the MXT keyword in the SIT controls the amount of space allocated by CICS for kernel stacks, and concurrently sets a limit on the maximum number of active tasks. Too high a value wastes storage and can cause CICS to become short-on-storage or experience other stress conditions. When executing in Goal Mode, a large MXT value can significantly increase overhead. CPExpert computes the percent of active tasks as a function of the value of the MXT keyword. When the percent of active tasks is less than the **PCTMXTLO** guidance variable, CPExpert concludes that the MXT value may be too high. Rule CIC104 is produced to report this conclusion.

If the condition exists for **any** CICS statistics interval in the data being analyzed for a region, CPExpert lists the percent of active tasks as a function of all intervals, and places ‘***’ beside those intervals in which the conclusion was reached. This listing is done to give a sense of perspective of (1) how often and (2) when the condition was detected.

Initially, CPExpert arbitrarily produced Rule CIC104 if the number of active tasks was less than 75% of the MXT value. The **PCTMXTLO** value was introduced to allow users to specify guidance to CPExpert. The value of the PCTMXTLO guidance variable can be specified either as “nn%” or as “.nn” to indicate a percentage.

Chapter 2.33: Region is approaching maximum capacity (PCTQRTCB)

CICS always has two or three TCBs for normal processing, depending on the release of CICS:

- The quasi-reentrant (**QR** mode) TCB executes the quasi-reentrant application code and most CICS code. This TCB is available in all releases of CICS.

- The resource-owning (**RO** mode) TCB is used for opening and closing data sets and for program loading. This TCB is available in all releases of CICS.

- The file-owning (**FO** mode) TCB is used for opening and closing data sets. This TCB is available beginning with CICS/Transaction Server for OS/390, Release 1.

CICS optionally has other TCBs, depending on the release of CICS.

If the Monitoring Class option is ON (MNPER="ON" in the System Initialization Table), the dispatcher domain maintains an "accumulated CPU TCB time" in the DSGACT variable.

Additionally, the dispatcher maintains "accumulated time dispatched" and "accumulated time in MVS wait" variables (DSGTDT and DSGTWT, respectively). The sum of the *accumulated time dispatched* and *accumulated time in MVS wait* is approximately the elapsed time that CICS was operational.

Dividing the *accumulated CPU TCB time* by the sum of the *accumulated time dispatched* and *accumulated time in MVS wait* yields an approximation of the **percent CPU busy** of the CICS region.

The DSGACT value does not include uncaptured CPU time, so the value normally will be less than the amount of CPU time actually used by the CICS region. IBM's *CICS Performance Guides* state that even with a totally busy CICS region, the calculated percent CPU busy of the CICS region would not normally be 100%. Consequently, IBM suggests that a region should be considered approaching maximum capacity if the calculated percent CPU busy exceeds 70%.

CPExpert produces Rule CIC109 when the Percent CPU Busy is greater than the value specified by the **PCTQRTCB** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTQRTCB** guidance variable is 60% indicating that CPExpert should produce Rule CIC109 whenever more than the Percent CPU Busy was more than 60% for the CICS region. The default guidance value is less than the 70% suggested by IBM to give an "early warning" of a capacity restraint. You can provide different guidance to CPExpert by changing the PCTQRTCB variable if you feel that Rule CIC109 is produced prematurely.

Chapter 2.34: Waits for IMS PSB pool space - PSBWAIT variable

The PSBPL operand in the SIT specifies the number of blocks in the CICS-DL/I program specification block (PSB) pool. The PSB pool is used only if the CICS-DL/I interface is used. CICS uses the value specified in the PSBPL operand to limit the total virtual storage allocated at any one time to the PSB pool. CICS does not reserve the amount of storage specified, but allocates and deallocates the storage as required.

If there is insufficient space in the PSB pool to handle PSB pool requests, an IMS routine is called to free the least-used buffers in the pool. In this case, the oldest PSB is deleted and the new PSB is loaded. The deletion (and particularly the loading) of PSBs slows the processing of tasks.

CPExpert produces Rule CIC194 if the number of waits for PSB pool space was greater than the **PSBWAIT** guidance variable, and if storage was not a constraint. The default specification for this guidance variable is **%LET PSBWAIT = 0**, indicating that Rule CIC194 will be produced if there were any waits for PSB pool space and storage was not constrained. You can provide different guidance to CPExpert by changing the PSBWAIT variable if you feel that Rule CIC194 is produced prematurely.

Chapter 2.35: Persistent Verification Timeouts - PVCOUNT variable

With CICS/ESA Release 3.2.1, "persistent verification" is the term used to describe signing on to a remote system and having that sign on remain valid (or "persist") over multiple conversations until it is no longer needed.

The PVDELAY parameter in the SIT is the "persistent verification" specification. The parameter specifies how long entries can remain signed on to the remote system when you are running remote transactions over ISC and IRC links.

Beginning with CICS/ESA Release 3.2.1, the CICS ISC/IRC Attach Time statistics provide the current value of the PVDELAY parameter (A21LUITM), provide the average time that has elapsed between each reuse of entries in the "signed on from" list (A21LUIAV), and provide the number of Persistent Verification (PV) timeouts (A21LUITI).

CPExpert produces Rule CIC255 if A21LUIAV is greater than A21LUITM, **and** A21LUITI is greater than the **PVCOUNT** guidance variable. The default specification for this variable is **%LET PVCOUNT = 0**, indicating that no persistent verification timeouts are acceptable. This low value is intended to alert you to the analysis process, and you probably will wish to change the value to a large number after your have reviewed the statistics. You can provide different guidance to CPExpert by changing the PVCOUNT variable if you feel that Rule CIC255 is produced prematurely.

Chapter 2.36: VTAM reached MAX RAPOOL - RAPOOL variable

Task input from a terminal is received by the VTAM access method and is passed to CICS, if CICS has a receive-any request outstanding. CICS receives data from VTAM in the receive-any input area (RAIA). The RAMAX operand in the SIT specifies the size in bytes of the I/O area that is allocated for each VTAM receive-any operation.

The RAPOOL operand in the System Initialization Table (SIT) specifies the number of concurrent receive-any requests that CICS is to process from VTAM. If RAPOOL is set too low, not all terminal messages may be processed during one dispatch of the terminal control program. This will cause transactions to be delayed and performance to suffer. If RAPOOL is set too high (and RAMAX is a large value), then storage is wasted.

CPExpert evaluates the "number of times reached maximum" in the VTAM statistics portion of the CICS requested or interval statistics. CPExpert produces Rule CIC150 if this value exceeds the **RAPOOL** guidance variable in USOURCE.CICGUIDE, and if storage was not constrained. The default specification for this guidance variable is **%LET RAPOOL = 0**. You can provide different guidance to CPExpert by changing the RAPOOL variable if you feel that Rule CIC150 is produced prematurely.

Chapter 2.37: SNT Timeouts - SNTCOUNT variable

The ISRDELAY parameter in the SIT is the "intersystem refresh delay" specification. The parameter specifies how long entries can remain signed on to the remote system when you are running remote transactions over ISC and IRC links.

Beginning with CICS/ESA Release 3.2.1, the CICS ISC/IRC Attach Time statistics provide the current value of the ISRDELAY parameter (A21SNTTM), provide the average time that has elapsed between each reuse of userids (A21SNTAV), and provide the number of Signon Table (SNT) timeouts (A21SNTTI).

CPExpert produces Rule CIC254 if A21SNTAV is greater than A21SNTTM, **and** A21SNTTI is greater than the **SNTCOUNT** guidance variable. The default specification for this variable is **%LET SNTCOUNT = 0**, indicating that no SNT timeouts are acceptable. This low value is intended to alert you to the analysis process, and you probably will wish to change the value to a large number after your have reviewed the statistics. You can provide different guidance to CPExpert by changing the SNTCOUNT variable if you feel that Rule CIC254 is produced prematurely.

Chapter 2.38: Number of storage dumps - STORDUMP variable

Storage dumps are produced for a variety of reasons (e.g., program checks and storage violations). The system is severely loaded while writing a dump to the CICS dump data

set. In the case of formatted dumps, nothing else is processed by CICS until the dump is written to the CICS dump data set. Depending upon the size of the dump, the dump could take over one second. During this time, all other activities are delayed.

CPExpert produces Rule CIC130 if the number of storage dumps is greater than the **STORDUMP** guidance variable in USOURCE.CICGUIDE. The default specification for this guidance variable is **%LET STORDUMP = 1**. You can provide different guidance to CPExpert by changing the STORDUMP variable if you feel that Rule CIC130 is produced prematurely.

Chapter 2.39: Waiting for VSAM Strings - STRWAIT variable

A VSAM "string" is a request to a VSAM data set requiring "positioning" within the data set. Each string results in a number of VSAM control blocks being built. If multiple strings are available, multiple VSAM requests can be executed concurrently. If multiple VSAM requests are executed concurrently, the buffers will be transferred to DASD quicker, since several I/O requests can be outstanding at one time.

The CICS interval statistics provide information about the number of times tasks were required to wait for strings. This information is provided for each VSAM file (summary information is provided for LSR pools).

CPExpert divides the wait-on-string values in the CICS interval statistics by the total file access operations. CPExpert produces Rule CIC160 if the resulting percentage is greater than the **STRWAIT** guidance variable. The default specification for this guidance variable is **%LET STRWAIT = 5**, indicating that Rule CIC160 should be produced if more than 5% of the VSAM file I/O operations wait for VSAM strings.

The CICS Performance Guide indicates that it may be acceptable to have 5% of the file accesses for NSR files waiting on strings, but no guidance is provided for LSR files. You can provide different guidance to CPExpert by changing the STRWAIT variable.

Chapter 2.40: Number of transaction errors - TRANSERR variable

Transaction errors are errors in which the transaction associated with a particular terminal could not be started. This means that:

- A transaction identifier had not been defined in the CICS System Definition (CSD) data set or in the Program Control Table (DFHPCT).
- The operator did not have the proper security access to enter the transaction.
- The transaction had been disabled.

CPExpert produces Rule CIC140 when the number of transaction errors for any particular terminal exceeds the **TRANSERR** guidance variable in USOURCE.CICGUIDE. The default specification for this guidance variable is **%LET TRANSERR = 0**. You can provide different guidance to CPExpert by changing the TRANSERR variable if you feel that Rule CIC140 is produced prematurely.

Chapter 2.41: Shipped terminal deletes - TRMSKDEL variable

In a transaction routing environment, terminal definitions can be "shipped" from a terminal-owning region (TOR) to an application-owning region (AOR). The terminal definition becomes redundant in the AOR under the certain conditions. Shipped terminal definitions which have become redundant may need to be deleted. The CICS-supplied transaction CRMF periodically scans the shipped terminal definitions in the AOR and flags those which it has determined to be redundant. The system initialization parameter DSHIPINT controls the frequency at which shipped terminal definitions are tested for redundancy. If the default value of 12 hours (or a similarly high value) is specified for the DSHIPINT parameter, the CRMF transaction might identify a considerable number of redundant terminal definitions during a single expiration of the DSHIPINT value. Consequently, a relatively large “burst” of CPU would be required for the CRMD transaction processing.

CPExpert compares the A04SKDEL variable in CICSAUTO (the number of shipped terminals deleted) with the **TRMSKDEL** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC156 when the number of shipped terminals deleted exceeds the value specified by the **TRMSKDEL** guidance variable.

The default value for the **TRMSKDEL** guidance value is 50, indicating that Rule CIC156 will be produced when more than 50 shipped terminal definitions have been deleted during any CICS interval statistics period. You can change this guidance if you fell that Rule CIC156 is produced prematurely.

Chapter 2.42: % TS I/O requests buffer wait - TSIOWAIT variable

VSAM multiple buffers allow multiple VSAM control intervals to be available in storage at the same time. When multiple buffers are available for temporary storage, the CICS temporary storage programs can use different buffers to respond to several concurrent requests for temporary storage. Using multiple VSAM buffers also increases the probability that the control interval required by any particular request will already be available in a buffer. If no buffer is available when a task attempts to access temporary storage, the task must wait. This situation is called "wait on buffers" and tasks are queued serially by temporary storage queue name.

It is generally desirable to minimize the number of times tasks must wait for access to a VSAM buffer. When the tasks wait, they continue to hold system resources (e.g., real storage), and may cause additional performance problems because the resources are not available to service other tasks.

CPExpert computes the percent of temporary storage I/O requests that waited for VSAM buffers by dividing the number of waits by the number of temporary storage I/O requests. CPExpert produces Rule CIC186 if this percent exceeds the **TSIOWAIT** guidance variable in USOURCE.CICGUIDE. The default specification for this guidance variable is **%LET TSIOWAIT = 5**, based upon guidelines in IBM's CICS Performance Guide. You can provide different guidance to CPExpert by changing the TSIOWAIT variable.

Chapter 3: Specifying guidance for individual CICS regions

This chapter applies only if you analyze multiple CICS regions during a single execution of the CICS Component of CPExpert and wish to provide unique guidance to certain CICS regions.

Some users of the CICS Component use CPExpert to analyze dozens or hundreds of CICS regions. These regions provide service to different users, the regions often have different applications, and the regions may even be a mixture of test regions and production regions. One set of analysis guidance variables do not always apply to all regions being analyzed. Consequently, CPExpert provides the capability to override the basic guidance on a region-by-region basis. This chapter describes how to implement this override guidance.

Unique guidance for particular CICS regions can be accomplished using the **GUIDE** guidance variable and the **applid_x = GUIDE** guidance variables, placed in the *ANALYSIS CONTROL VARIABLES* section of USOURCE(CICGUIDE), and placing the unique guidance into a USOURCE(applid_x) PDS member. Exhibit 3-2 describes the *ANALYSIS CONTROL VARIABLES* section.

Chapter 3.1: Specify GUIDE guidance variable

The **GUIDE** guidance variable specifies whether override guidance exists for one or more regions.

CPExpert normally uses the guidance in USOURCE(CICGUIDE) to control its analysis of CICS regions. You can override this process by specifying **%LET GUIDE = Y;** in the *ANALYSIS CONTROL VARIABLES* section of USOURCE(CICGUIDE). When you make this specification, CPExpert generates code to acquire additional guidance for particular CICS regions, as controlled by the **applid_x = GUIDE** guidance variable(s).

The purpose of having a “general” controlling parameter is to provide a “switch” that allows you to normally use override guidance, but you can easily turn off the overrides and analyze all regions using the basic guidance.

Chapter 3.2: Identify specific CICS regions having unique guidance

Each CICS region for which you wish to provide specific guidance must be identified to CPExpert. This identification is done by specifying **%LET applid_x = GUIDE;** in USOURCE(CICGUIDE). For example, if you wish to specify unique guidance for APPLID CICSPVA7, you would specify **%LET CICSPVA7=GUIDE;** in USOURCE(CICGUIDE). This specification tells CPExpert that unique guidance for APPLID CICSPVA7 is contained in USOURCE(CICSPVA7).

Chapter 3.3: Place unique guidance in USOURCE(applid) member

Create a unique analysis guidance member in the USOURCE partitioned data set. For example, if you wish to specify unique guidance for APPLID CICSPVA7, you would create a USOURCE(CICSPVA7) member.

You can specify any unique guidance for the particular APPLID by entering the appropriate SAS %LET statements into the USOURCE(applid) member. For example, if you wish to specify unique guidance for APPLID CICSPVA7, you would specify appropriate SAS %LET statements in USOURCE(CICSPVA7).

Important: If you should simply copy the USOURCE(CICGUIDE) member to USOURCE(applid) member, you must delete from the new module all information unrelated to the unique guidance. **Do not retain data selection and reporting variables**! Unpredictable analysis could result if you retain the data selection and reporting variables.

Chapter 3.4: Restrictions

The following restrictions apply to specifying unique guidance for CICS regions when processing multiple CICS regions during a single execution of the CICS Component:

- You can specify unique guidance for up to 99 CICS regions.
- The override guidance specific to a particular CICS region applies only to that region. CPExpert reverts to the original guidance contained in USOURCE(CICGUIDE) for subsequent CICS regions (unless those CICS regions have unique guidance).
- You will get a SAS error if you specify %LET applid_x=GUIDE; and do not have the corresponding member as USOURCE(applid_x).
- You can “comment out” any %LET applid_x=GUIDE; statement simply by replacing the “%” with “*” (that is, replace the percent with an asterisk)..

Chapter 4: Analyzing CICS report classes

Some users of the CICS Component want CPExpert to provide a "warning" about certain system-level resource use or performance characteristics related to specific CICS workload. To meet this requirement, the CICS Component has been enhanced to provide the ability to associate specific CICS workload with system-level information contained in RMF Type72 records.

The initial requirement was for CPExpert to provide a warning when certain CICS workload used more than a specified percent of CPU[4]. For this requirement, logic was implemented to associate specific CICS workload with the MXG TYPE72GO information, and guidance was implemented using the CPUWARN guidance variable. While other features may be implemented, the "CPU warning" construct will be used to discuss and illustrate both the concept and the implementation of the enhancement.

The "CPU warning" approach requires that a Report Class describing the CICS workload of interest be defined to the Workload Manger (WLM). This Report Class is defined using the standard WLM workload classification panel.

During CICS operation, the System Resources Manager (SRM) acquires information related to the defined Report Class, and RMF writes the Report Class information[5] to SMF as part of the SMF Type 72 records. MXG processes the SMF Type 72 records and places the SMF information into various files in the MXG performance data base. CPExpert processes the MXG TYPE72GO file to extract information about the defined Report Class. CPExpert produces a "CPU warning" when the defined Report Class uses more than "nn%" CPU during any RMF reporting interval.

Chapter 4.1: Implementing analysis of CICS report classes

A user of the CICS Component implements the Report Class analysis feature by the following steps:

[4] This feature is similar to the analysis described in Rule CIC109; however, Rule CIC109 relates to the quasi-reentrant (QR) TCB time, which is considered by IBM to be not as representative of the CPU time required by the region as would be data produced by RMF. For a warning regarding CPU time used by a region, either the "QR TCB approach" described in Rule CIC109 or the "report class approach" described by this discussion probably would yield similar results.

[5] The data available for report classes include:

- Number of transactions completed
- Average response times
- Resource usage data
- State samples.

Additional data was added to report class information, beginning with z/OS Version 1, Release 2:

- Response time distribution buckets.
- Work manager delay data.

- **Step 1. Define Report Class**. The "Defining Report Classes" section of IBM's *MVS Planning: Workload Management* document describes how to define report classes to the WLM. In brief, the WLM's classification rules can assign incoming work to a report class. Report classes can be used to report on a subset of transactions running in a single service class, or can can be used to report on a combination of transactions running in different service classes within one report class.

- **Step 2: Specify that report classes are to be analyzed by the CICS Component**. The **RPRTCLASS** guidance variable is used to tell CPExpert that report classes are to be analyzed. If you wish the CICS Component to analyze report classes, specify **%LET RPTCLASS=Y;** in USOURCE(CICGUIDE).

- **Step 3: Identify the report classes to be analyzed**. There can be many report classes defined using the WLM Report Class Definition panel. You must tell CPExpert which specific report classes are to be analyzed by the CICS Component. This is done by specifying **%LET RPRTn=report_class_name;** in USOURCE(CICGUIDE). For example, to tell CPExpert that the CICSAOR3 report class should be analyzed, specify **%LET RPRT1=CICSAOR3;** in USOURCE(CICGUIDE). You can specify up to 99 report classes[6] to be analyzed by the CICS Component.

Once the above steps have been completed, the CICS Component will process MXG TYPE72GO, selecting records that meet the overall data selection criteria (SYSTEM, start/end dates and times, etc.), and that are for the defined report classes. Subsequent analysis will apply the appropriate evaluation logic (e.g., when the CICS workload identified by the report classes used more than the percent of CPU specified by the CPUWARN guidance variable). The CIC700(series) rules will be produced when the resource use exceeds the guidance.

Chapter 4.2: Specifying guidance for individual CICS report classes

The CICS Component can be used to analyze resource use or performance characteristics of multiple CICS report classes during a single execution of the CICS Component. One set of analysis guidance variables does not always apply to all CICS report classes being analyzed. Consequently, CPExpert provides the capability to override the basic guidance on a report-class-by-report-class basis. This chapter describes how to implement this override guidance.

Chapter 4.2.1: Specify RPTGUIDE guidance variable

The **RPTGUIDE** guidance variable specifies whether override guidance exists for one or more report classes.

[6] Please notify **Don_Deese@cpexpert.com** if you require more than 99 report classes.

CPExpert normally uses the guidance in USOURCE(CICGUIDE) to control its analysis of CICS report classes. You can override this process by specifying **%LET RPTGUIDE = Y;** in the *ANALYSIS CONTROL VARIABLES* section of USOURCE(CICGUIDE). When you make this specification, CPExpert generates code to acquire additional guidance for particular CICS regions, as controlled by the **report_class_x = GUIDE** guidance variable(s).

The purpose of having a "general" controlling parameter is to provide a "switch" that allows you to normally use override guidance, but you can easily turn off the overrides and analyze all regions using the basic guidance.

Chapter 4.2.2: Identify specific report classes having unique guidance

Each CICS report class for which you wish to provide specific guidance must be identified to CPExpert. This identification is done by specifying **%LET report_class_x = GUIDE;** in USOURCE(CICGUIDE). For example, if you wish to specify unique guidance for CICS Report Class CICSAOR3, you would specify **%LET CICSAOR3=GUIDE;** in USOURCE(CICGUIDE). This specification tells CPExpert that unique guidance for Report Class CICSAOR3 is contained in USOURCE(CICSAOR3).

Chapter 4.2.3: Place guidance in USOURCE(report_class) member

Create a unique analysis guidance member in the USOURCE partitioned data set. For example, if you wish to specify unique guidance for Report Class CICSAOR3, you would create a USOURCE(CICSAOR3) member.

You can specify any unique guidance for the particular CICS report class by entering the appropriate SAS %LET statements into the USOURCE(report_class) member. For example, if you wish to specify unique guidance for Report Class CICSAOR3, you would specify appropriate SAS %LET statements in USOURCE(CICSAOR3).

Important: If you should simply copy the USOURCE(CICGUIDE) member to USOURCE(applid) member, you must delete from the new module all information unrelated to the unique guidance. **Do not retain data selection and reporting variables**! Unpredictable analysis could result if you retain the data selection and reporting variables.

Chapter 4.2.4: Restrictions

The following restrictions apply to specifying unique guidance for CICS report classes when processing multiple CICS report classes during a single execution of the CICS Component:

- You can specify unique guidance for up to 99 CICS report classes.

- The override guidance specific to a particular CICS report class applies only to that report class. CPExpert reverts to the original guidance contained in USOURCE(CICGUIDE) for subsequent CICS report classes (unless those CICS report classes have unique guidance).

- You will get a SAS error if you specify %LET report_class_x=GUIDE; and do not have the corresponding member as USOURCE(report_class_x).

- You can "comment out" any %LET report_class_x=GUIDE; statement simply by replacing the "%" with "*" (that is, replace the percent with an asterisk).

Chapter 5: System Logger Analysis Guidance Variables

The system logger is an MVS component that allows an application to log data from a sysplex. The system logger component resides in its own address space on each system in a sysplex. Applications can log data from one system or from multiple systems across the sysplex.

Applications write log data into a *log stream*. From the MVS view, the log stream is a set of records in time sequence order, merged into a single stream, independent of physical residence of the log stream. The log stream can reside in data space storage, in a staging data set, in a coupling facility, or in a log stream DASD data set. System parameters control the placement and length of log stream.

Applications that use the system logger services include:

- **Logrec**. Logrec log stream is an MVS system logger application that records hardware failures, selected software errors, and selected system conditions across the sysplex.

- **Operations log (OPERLOG)**. OPERLOG is an MVS system logger application that records and merges messages about programs and system functions (the hard copy message set) from each system in a sysplex that activates OPERLOG.

- **CICS Log Manager with CICS/Transaction Server for OS/390**. CICS log manager is a CICS system logger application that replaces the journal control management function.

- **IMS Common Queue Server Log Manager**. IMS common shared queues (CQS) log manager is a system logger application that records the information necessary for CQS to recover structures and restart.

- **APPC/MVS**. APPC/MVS is an MVS system logger application that records events related to protected conversations.

- **RRS (resource recovery services)**. RRS is an MVS system logger application that records events related to protected resources.

One significant advantage of the MVS system logger design is that any other system in a sysplex can recover data in the log stream. This feature prevents data loss in case of failure of one system.

Prior to OS/390 Release 2.4, the MVS system logger required a coupling facility (unless appropriate APARs were installed with OS/390 Release 1.3). With OS/390 Release 1.3 (or OS/390 Release 1.3 with appropriate APARs), individual log streams can use either DASD or a coupling facility.

Data in a log stream is contained in two kinds of storage: (1) *interim storage*, where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is "hardened" for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is off loaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD_only log streams. The main difference between the two types of log streams is the storage medium system logger uses to hold interim log data:

- In a coupling facility log stream, interim storage is coupling facility list structures.

- In a DASD_only log stream, interim storage is contained in local storage buffers on the system, as an MVS data space areas associated with the system logger address space.

Additionally, for data integrity there exists duplexed storage, so that if one system or component fails, the log stream can be recovered from the duplexed storage. These concepts differ, depending on whether the log stream is defined for a coupling facility or for DASD-only.

- If the primary storage is defined as a list structure in a coupling facility, the duplexed data can be retained in another coupling facility, or can be retained in *staging data sets*. Staging data sets are used when the coupling facility is in the same CPC, or uses volatile storage.

- If the primary storage is defined as DASD-only, the duplexed data is retained in *staging data sets*.

Interim storage normally is "offloaded" to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage[7] being filled. For log streams defined in coupling facility list structures, the parameters apply to the coupling facility structures[8]. For log streams defined as DASD-only, these parameters apply to the log stream staging data set.

Once log stream data has been offloaded, the MVS system logger releases the storage in the list structure or staging data set, so the space can be used to hold new log blocks. From an application point of view, the actual location of the log data in the log stream is transparent.

[7] The controls apply **only** to staging data set usage with DASD-only log streams. With coupling facility log streams, the controls apply to both coupling facility structure usage and staging data set usage if the log stream is duplexed to staging data sets.

[8] The parameters will also apply to staging data sets if the log stream is duplexed to staging data sets.

CPExpert's system logger analysis guidance variables allows you to guide CPExpert's analysis logger performance problems. Default thresholds have been established based on information contained in IBM publications and other documents. These defaults may not be suitable for your environment and specific management objectives. If the analysis and reports produced by CPExpert do not meet your needs, alter the guidance to CPExpert. If the guidance is insufficient, please call Computer Management Sciences at (703) 922-7027 so we can make changes to improve CPExpert for you!

Exhibit 3-3 illustrates the USOURCE(CICGUIDE) variables that provide guidance to the CICS Component as it analyzes system logger performance. This chapter describes these variables, how the variables are used, and how the variables are altered.

```
*********************************************************************   ;
*       CICS Component GUIDANCE VARIABLES                             ;
*********************************************************************   ;
       •
       •
       •
       •

**** SYSTEM LOGGER GUIDANCE VARIABLES                                    ;
%LET SMFTYP88 =N        ; * TYPE 88 RECORDS AVAILABLE IN MXG?           ;
%LET LGDSFULL =0        ; * ACCEPTABLE LOG STREAM STAGING DATA SET FULL;
%LET LGSHIFTS =1        ; * ACCEPTABLE NUMBER OF LOG STREAM DASD SHIFTS;
%LET PCTINST  =0        ; * PERCENT INTERIM STORAGE NOT EFFECTIVELY USE;
%LET PCTLOCST =0        ; * PERCENT LOCAL STORAGE NOT EFFECTIVELY USED ;
%LET STDSHIGH =0        ; * STAGING DATA SET HIGH THRESHOLD HIT         ;
%LET STFULL90 =0        ; * ACCEPTABLE STRUCTURE 90% FULL               ;
%LET STRC2    =0        ; * ACCEPTABLE TIMES STRUCTURE HIGHOFFLOAD HIT ;
%LET STRC3    =0        ; * ACCEPTABLE TIMES STRUCTURE CRITICAL SPACE  ;
%LET STRFULL  =0        ; * ACCEPTABLE LOG STREAM STRUCTURE FULL        ;

%LET LOGGUIDE = N       ; * GUIDANCE PROVIDED FOR INDIVIDUAL LOG STRMS ;
/* SPECIFY GUIDANCE FOR LOG STREAMS: EXAMPLE
LOGNAME = log.stream.name1
%LET PCTINST  =0        ; * PERCENT INTERIM STORAGE NOT EFFECTIVELY USE;
%LET PCTLOCST =0        ; * PERCENT LOCAL STORAGE NOT EFFECTIVELY USED ;
%LET STFULL90 =0        ; * ACCEPTABLE STRUCTURE 90% FULL               ;
%LET STDSHIGH =0        ; * STAGING DATA SET HIGH THRESHOLD HIT         ;
*/
```

DEFAULT VALUES FOR SYSTEM LOGGER ANALYSIS

EXHIBIT 3-3

Chapter 5.1: SMF Type 88 records available - SMFTYP88 variable

The SMFTYP88 guidance variable is used to tell CPExpert whether you wish to analyze system logger performance based on SMF Type 88 records. If the SMFTYP88 variable is **N**, CPExpert will **not** analyze system logger performance based on SMF Type 88 records. If the SMFTYP88 variable is **Y**, CPExpert **will** analyze system logger performance based on SMF Type 88 records.

System logger analysis applies only with a MXG (or SAS/ITSV) performance data base, as CA-MICS does not provide a SAS file contain the SMF Type 88 variables.

Chapter 5.2: Log stream staging data set full - LGDSFULL variable

For a DASD-only log stream or for a log stream that is duplexed to a staging data set, a 'STAGING DATA SET FULL' condition can exist. In this case, the staging data set has reached its capacity before off loading data to secondary storage. Once the staging data set space for a log stream is filled, system logger rejects all write requests until the staging data set log data can be offloaded to DASD log data sets.

This situation can cause the application to wait before it can write more data. Depending on the length of time the application must wait, significant performance degradation would be experienced.

CPExpert compares the SMF88ETF (times a staging data set full was detected) variable in the MXG TYPE88 data set with the **LGDSFULL** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC302 when the SMF88ETF value exceeds the **LGDSFULL** guidance variable.

The default value for the **LGDSFULL** guidance variable is zero, indicating that CPExpert should produce Rule CIC302 when any staging data set full condition was detected. You can alter this analysis using the **LGSDSFULL** guidance variable. For example, if you wish to be notified only when the staging data set for the log stream is full more than 5 times in an SMF interval, specify:

```
%LET LGDSFULL = 5 ; * ACCEPTABLE LOG STREAM STAGING DATA SET FULL;
```

From a practical matter, you should always wish to be notified that a log stream staging data set experienced a “FULL” condition. Please notify Computer Management Sciences at (703) 922-7027 if you have a situation where you wish to modify this guidance variable (so we can better appreciate unique situations).

Chapter 5.3: Log stream DASD-shift conditions - LGSHIFTS variable

A log stream can have data in multiple DASD log data sets. As an offload data set becomes full, the system logger automatically allocates a new one for the log stream. This process is known as a "DASD-shift" and *generates considerable overhead*. Consequently, a "DASD-shift" should not occur frequently. IBM suggests that "DASD-shifts" should occur no more than once per hour.

CPExpert examines the SMF88EDS variable (the number of log stream DASD shifts during the SMF interval). While IBM suggests that you not have more than one DASD shift per hour, an SMF recording interval typically is less than an hour (normally the interval is 15 minutes). Consequently, CPExpert calculates the number of SMF intervals in an hour and tracks the number of DASD shifts that occur during any hour.

CPExpert produces Rule CIC307 when the number of DASD shifts that occur during any hour exceeds the **LGSHIFTS** guidance variable in USOURCE(CICGUIDE). The default value for the **LGSHIFTS** is one, indicating that CPExpert should produce Rule CIC307 when more than one log stream DASD shift occurred during any hour. You can alter this analysis using the **LGSDSFULL** guidance variable. For example, if you wish to be notified only when more than two DASD shifts occur in one hour, specify:

```
%LET LGSHIFTS = 2 ; * ACCEPTABLE NUMBER OF LOG STREAM DASD SHIFTS;
```

Chapter 5.2: Percent interim storage offloaded - PCTINTST variable

Data in a log stream is contained in two kinds of storage: (1) *interim storage*[9], where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is "hardened" for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

Interim storage normally is "offloaded" to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled.

Once log stream data has been offloaded, the MVS system logger releases the storage in the list structure, so the space in the structure can be used to hold new log blocks.

[9] Interim storage is sometimes referred to as "primary" storage.

From an application point of view, the actual location of the log data in the log stream is transparent.

Applications using system logger services (such as CICS/Transaction Server for OS/390) often manage the system log by deleting records for completed units of work during activity keypoint processing (this is also called log-tail deletion). The number of bytes deleted from the system log after writing to offload data sets should be very low. Unnecessary overhead is incurred when data is moved to the offload data sets, only to be later deleted. With an appropriately sized log stream, the system log data remains in interim storage, and the overhead of data spilling to DASD simply to be deleted later is avoided.

CPExpert computes the percent of ineffective use of interim storage (PCTINTST) by applying the following algorithm:

$$PCTINTST = \frac{SMF88SAB}{SMF88SIB + SMF88SAB}$$

where:

SMF88SAB = Bytes deleted after being offloaded
SMF88SIB = Bytes deleted before being offloaded

CPExpert compares the computed PCTINTST with the **PCTINTST** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC304 when the percent ineffective use of use of interim storage exceeds the value specified by the **PCTINTST** guidance variable.

The default value for the **PCTINTST** guidance variable is zero, indicating that CPExpert should produce Rule CIC304 whenever interim storage use was not effective. You can alter this analysis using the **PCTINTST** guidance variable. For example, if you wish to be notified only when more than ten percent of the bytes were deleted after offload, specify:

```
%LET PCTINTST = 10% ; * PERCENT INTERIM STORAGE NOT EFFECTIVELY USED;
```

Chapter 5.3: Percent use of staging data sets - PCTLOCT variable

Data in a log stream is contained in two kinds of storage: (1) *interim storage*[10], where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*,

[10] Interim storage is sometimes referred to as “primary” storage.

where data is "hardened" for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD-only log streams. The main difference between the two types of log streams is the storage medium that the system logger uses to hold interim log data:

- With a coupling facility log stream, interim storage is contained in coupling facility list structures. The system logger duplexes the log stream to either (1) MVS data space areas associated with the system logger address space or (2) staging data sets, depending on whether the coupling facility is failure-independent. Interim storage residing in coupling facility structures is analyzed by Rule CIC304 and the analysis is guided by the PCTINTST guidance variable.

- With a DASD-only log stream, interim storage is contained in local storage buffers on the system (as MVS data space areas associated with the system logger address space). With a DASD-only log stream the system logger duplexes the log stream to staging data sets. Interim storage residing in local storage buffers is analyzed by Rule CIC305 and the analysis is guided by the PCTLOCST guidance variable.

Interim storage normally is "offloaded" to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled.

For log streams defined in coupling facility list structures, the parameters apply to the coupling facility structures[11]. For log streams defined as DASD-only, these parameters apply to the **log stream staging data set**.

Once log stream data has been offloaded, the MVS system logger releases the storage in the staging data sets, so the space in the staging data sets can be used to hold new log blocks. From an application point of view, the actual location of the log data in the log stream is transparent.

Applications using system logger services (such as CICS/Transaction Server for OS/390) often manage the system log by deleting records for completed units of work during activity keypoint processing (this is also called log-tail deletion). The number of bytes deleted from the system log after writing to offload data sets should be very low. Unnecessary overhead is incurred when data is moved to the offload data sets, only to be later deleted. With an appropriately sized log stream, the system log data remains in interim storage, and the overhead of data spilling to DASD simply to be deleted later is avoided.

[11] The parameters will also apply to staging data sets if the log stream is duplexed to staging data sets. Problems with staging data set threshold being encountered are analyzed in Rule CIC305.

CPExpert computes the percent of ineffective use of staging data sets (PCTLOCST) by applying the following algorithm to DASD-only log streams

$$PCTLOCST = \frac{SMF88SAB}{SMF88SIB + SMF88SAB}$$

where

SMF88SAB = Bytes deleted after being offloaded

SMF88SIB = Bytes deleted before being offloaded

CPExpert compares the computed PCTLOCST with the **PCTLOCST** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC305 when the percent ineffective use of use of interim storage exceeds the value specified by the **PCTLOCST** guidance variable.

The default value for the **PCTLOCST** guidance variable is 0, indicating that CPExpert should produce Rule CIC305 whenever DASD staging data set use was not effective. You can alter this analysis using the **PCTLOCST** guidance variable. For example, if you wish to be notified only when more than ten percent of the local storage bytes were deleted after offload, specify:

```
%LET PCTLOCST = 10% ; * PERCENT LOCAL STORAGE NOT EFFECTIVELY USED;
```

Chapter 5.4: Staging data set threshold - STDSHIGH variable

Interim storage in a coupling facility structure normally is “offloaded” to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled.

Additionally, for data integrity there exists duplexed storage, so that if one system or component fails, the log stream can be recovered from the duplexed storage. These concepts differ, depending on whether the log stream is defined for a coupling facility or for DASD-only.

- If the primary storage is defined as a list structure in a coupling facility, the duplexed data can be retained in another coupling facility, or can be retained in *staging data sets*. Staging data sets are used when the coupling facility is in the same CPC, or uses volatile storage.

- If the primary storage is defined as DASD-only, the duplexed data is retained in *staging data sets*.

When a log stream in a coupling facility is duplexed to staging data sets, the system logger automatically makes a duplicate copy of the data every time data is written to a log stream. This is done to protect against data loss due to coupling facility problems or due to system failure. The duplicate copy is kept in the staging data sets until the data is off-loaded from the coupling facility structure to DASD log data sets. After the data is off-loaded to DASD log data sets, the system logger discards the duplicate copy of the log data.

Interim storage in a coupling facility structure normally is "offloaded" to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled. For log streams defined in coupling facility list structures, these parameters apply as follows:

- When the coupling facility structure is filled to the **high offload threshold** point or beyond, the system logger begins offloading data from the coupling facility to the DASD log stream data sets. For example, if the HIGHOFFLOAD parameter is specified as 80% (this is the default value), the system logger normally would begin offloading interim storage to DASD log data sets when 80% or more of the structure is used.

- The **low offload threshold** is the point in the coupling facility structure, as a percent space consumed , where the system logger stops offloading coupling facility log data to log stream DASD data sets. The default LOWOFFLOAD parameter value is 0%, indicating that the system logger will offload all the log stream to DASD log data sets once offloading has commenced.

For log streams in a coupling facility that are duplexed to staging data sets, the values of the HIGHOFFLOAD and LOWOFFLOAD parameters **apply to the staging data sets** as well as to the coupling facility structure. This is simply because if the staging data sets become full, MVS would not be able to continue duplexing data and there would be a data integrity exposure in case of failure. Consequently, if a staging data set fills up **before** an offload of a log stream in a coupling facility structure is triggered by the high threshold specification, an offload will be triggered because of the full staging data set.

When a staging data set reaches the high threshold, the system logger immediately offloads data from the coupling facility to DASD log data sets, even if the coupling facility usage for the log stream is below the high threshold. Thus, if the staging data sets are small in comparison to the coupling facility structure size for a log stream, the staging data sets will keep filling up and the system logger will frequently offload coupling facility data

to DASD log data sets. This means that your installation would experiences frequent (and unexpected) offloading overhead that could affect performance[12].

CPExpert examines the SMF88STN variable (the structure name) in the MXG TYPE88 data set to select records that apply only to coupling facility structures[13]. For these records, CPExpert examines the SMF88ETF variable (the number of times the system logger detected a Staging Data Set Threshold hit condition). CPExpert produces Rule CIC306 when the SMF88ETF value exceeds the **STDSHIGH** guidance variable in USOURCE(CICGUIDE).

The default value for the **STDSHIGH** is zero, indicating that CPExpert should produce Rule CIC306 whenever a Staging Data Set Threshold was encountered. You can alter this analysis using the **STDSHIGH** guidance variable. For example, if you wish to be notified only when the HIGHOFFLOAD threshold was encountered more than 5 times during an SMF recording interval, specify:

```
%LET STDSHIGH = 5 ; * TIMES STAGING DATA SET HIGH THRESHOLD HIT;
```

Chapter 5.5: Log stream structure 90% full - STFULL90 variable

When a coupling facility structure is defined, it is divided into two areas: One area holds *list elements*, and the other area holds *list entries*. List elements are units of logged data and are either 256 bytes or 512 bytes long. There is at least one element per log record. List entries are index pointers to the list elements. There is one list entry per log record.

Each log record places an entry in the list entry area of the structure, and the data is loaded as one or more elements in the list element area. **If the list entry area exceeds 90% of its capacity, all log streams are offloaded to DASD.** DASD offloading commences at this point, regardless of the current utilization of the log stream, and continues until an amount of data equal to the difference between the HIGHOFFLOAD threshold and the LOWOFFLOAD threshold for the log stream has been offloaded.

This situation can occur when log steams share a structure, one log stream is used by an application issuing very few journal write requests, and other applications issue frequent journal write requests to log streams in the same structure. All log streams may be offloaded to DASD because of the frequent journal write requests by the other applications.

[12] If your staging data sets are too small, you also run the risk of filling them up completely. If this occurs, system logger immediately begins offloading the coupling facility log data in DASD log data sets to harden it. System logger applications will be unable to log data until system logger can free up staging data set space. This serious situation is evaluated by Rule CIC302.

[13] The SMF88STN variable will be *DASDONLY* for log streams that are DASD-only log streams.

The primary disadvantage of encountering this situation is that the application that is infrequently writing to the log stream might not have its LOWOFFLOAD and HIGHOFFLOAD thresholds controlling the offload process. This can result in unpredictable offloading, and possibly undesirable performance.

For example, Log Stream A might have a HIGHOFFLOAD threshold of 80% and a LOWOFFLOAD threshold of 60%. Because of log stream activity by other applications writing to other log streams, the list entry area may exceed 90% of its capacity even though Log Stream A might be only 50% utilized. Although Log Stream A had not reached its HIGHOFFLOAD threshold, or even its LOWOFFLOAD threshold, data would be offloaded until 20% of the log stream was offloaded. This is the difference between 80% and 60%. After the offloading operation has completed, log stream A is at 30% utilization (50% minus 20%).

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file.

CPExpert examines the SMF88STN variable (the structure name) to select information that applies only to coupling facility structures[14]. For these records, CPExpert examines the SMF88EFS variable (offloads for all log streams connected from this system to this structure because structure was 90% full) in the SMF Type 88 records. CPExpert produces Rule CIC303 when the SMF88EFS value exceeds the **STFULL90** guidance variable in USOURCE(CICGUIDE).

The default value for the **STFULL90** is zero. Any non-zero value in the SMF88EFS variable indicates that the entry to element ratio is too high for the structure. You can alter this analysis using the **STFULL90** guidance variable. For example, if you wish to be notified only when offloads were scheduled because the structure was 90% full more than 5 times during an SMF recording interval, specify:

```
%LET STFULL90 = 5 ; * TIMES OFFLOADS BECAUSE STRUCTURE WAS 90% FULL;
```

Chapter 5.6: Log stream Type-2 completions - STRC2 variable

When a system logger user issues the IXGWRITE macro for a coupling facility log stream, the system logger writes to the coupling facility structure. When the write completes, the system logger categorizes the event as a *Type-1*, *Type-2*, or *Type-3* completion. The categorization indicates how much space in the structure is being used by the log stream when the completion occurred.

[14] The SMF88STN variable will be *DASDONLY* for log streams that are DASD-only log streams.

- A *Type-1* completion indicates that, after the write completed, the percentage of the structure space used was less than the HIGHOFFLOAD threshold, meaning that system logger is using the coupling facility successfully. This is a desired completion status.

- A *Type-2* completion indicates that, after the write completed, the percentage of the structure space used was equal to or greater than the HIGHOFFLOAD threshold. This means that the system logger begins managing storage resources by migrating data from the coupling facility to DASD log data sets.

- A *Type-3* completion indicates that a given log stream is close to consuming all the space in the coupling facility. A Type-3 completion can occur if there is a failure which prevents the system logger from promptly moving data from the coupling facility structure to DASD log data sets or if the system logger configuration is tuned incorrectly. The Type-3 completions are analyzed by Rule CIC309.

CPExpert examines the SMF88SC2 variable (Count of Type-2 completions) in the SMF Type 88 records. CPExpert produces Rule CIC308 when the SMF88SC2 value exceeds the **STRC2** guidance variable in USOURCE(CICGUIDE).

The default value for the **STRC2** is zero, indicating that CPExpert should produce Rule CIC308 whenever the HIGHOFFLOAD threshold was reached in an SMF interval. You can alter this analysis using the **STRC2** guidance variable. For example, if you wish to be notified only when the structure reached the HIGHOFFLOAD threshold more 5 times during an SMF recording interval, specify:

```
%LET STRC2 = 5 ; * TIMES STRUCTURE HIGHOFFLOAD THRESHOLD WAS HIT;
```

Chapter 5.7: Log stream Type-3 completions - STRC3 variable

When a system logger user issues the IXGWRITE macro for a coupling facility log stream, the system logger writes to the coupling facility structure. When the write completes, the system logger categorizes the event as a *Type-1*, *Type-2*, or *Type-3* completion. The categorization indicates how much space in the structure is being used by the log stream when the completion occurred.

- A *Type-1* completion indicates that, after the write completed, the percentage of the structure space used was less than the HIGHOFFLOAD threshold, meaning that system logger is using the coupling facility successfully. This is a desired completion status.

- A *Type-2* completion indicates that, after the write completed, the percentage of the structure space used was equal to or greater than the HIGHOFFLOAD threshold. This means that the system logger begins managing storage resources by migrating data from the coupling facility to DASD log data sets. The Type-2 completions are analyzed by Rule CIC308.

- A *Type-3* completion indicates that a given log stream is close to consuming all the space in the coupling facility. A Type-3 completion can occur if there is a failure which prevents the system logger from promptly moving data from the coupling facility structure to DASD log data sets or if the system logger configuration is tuned incorrectly. If a log stream has a large proportion of Type-3 completions, the system logger is getting dangerously close to the STRUCTURE FULL condition.

CPExpert examines the SMF88SC3 variable (Count of Type-3 completions) in the SMF Type 88 records. CPExpert produces Rule CIC309 when the SMF88SC3 value exceeds the **STRC3** guidance variable in USOURCE(CICGUIDE).

The default value for the **STRC3** is zero, indicating that CPExpert should produce Rule CIC309 whenever the space used by a log stream caused the coupling facility structure to reach a critical amount. You can alter this analysis using the **STRC3** guidance variable. For example, if you wish to be notified only when the log stream caused the structure to reach a critical amount more than 5 times during an SMF recording interval, specify:

```
%LET STRC3 = 5 ; * TIMES STRUCTURE CRITICAL SPACE  WAS HIT;
```

Chapter 5.8: Log stream CF structure full - STRFULL variable

Prior to OS/390 Release 2.4, the MVS system logger required a coupling facility (unless appropriate APARs were installed with OS/390 Release 1.3). With OS/390 Release 2.4 (or OS/390 Release 1.3 with appropriate APARs), individual log streams can use either DASD or a coupling facility.

- For a log stream that uses a coupling facility structure, a 'STRUCTURE FULL' condition can exist. In this case, the coupling facility has reached its capacity before off loading data to DASD[15]. This condition is analyzed by Rule CIC301.

- For a DASD-only log stream or for a log stream that is duplexed to a staging data set, a 'STAGING DATA SET FULL' condition can exist. In this case, the staging data set

[15]This condition could be encountered during the rebuilding of a coupling facility structure, but rebuilding of a coupling facility structure is an event that would not require CPExpert's analysis - such an event would be well-known to systems personnel!

has reached its capacity before off loading data to secondary storage. This condition is analyzed by Rule CIC302.

If either of the above situations occur, they indicate that the logger cannot write data to secondary storage quickly enough to keep up with incoming data. Once the coupling facility space for a log stream is filled, system logger rejects all write requests until the coupling facility log data can be offloaded to DASD log data sets. Both situations can cause the application to wait before it can write more data. Depending on the length of time the application must wait, significant performance degradation would be experienced.

CPExpert examines the SMF88STN variable in the MXG TYPE88 data set (this variable indicates whether the log stream is a coupling facility type, or is a DASDONLY type). When this variable indicates the log stream is a coupling facility type, CPExpert compares the SMF88ESF (times a structure full condition was detected) variable in the MXG TYPE88 data set with the **STRFULL** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC301 when the SMF88ESF value exceeds the **STRFULL** guidance variable.

The default value for the **STRFULL** guidance variable is zero, indicating that CPExpert should produce Rule CIC301 when any structure full condition was detected. You can alter this analysis using the **STRFULL** guidance variable. For example, if you wish to be notified only when a structure full condition was encountered more than 5 times during an SMF recording interval, specify:

```
%LET STRFULL = 5 ; * TIMES STRUCTURE FULL WAS ENCOUNTERED;
```

Chapter 5.9: Specifying guidance for specific log streams

The guidance variables for log streams as described above are globally applied during CPExpert's analysis of system logger performance from the perspective of CICS. These global guidance variables might not be applicable to some log streams, however. Guidance can **optionally** be applied to specific log streams or to the coupling facility structures used by the log streams.

Guidance for specific log streams or log stream structures is accomplished by specifying **%LET LOGGUIDE=Y;** in USOURCE(CICGUIDE), identifying the log stream(s) or log stream structure(s) to which the guidance applies, and specifying guidance variables for the log stream(s) or log stream structure(s).

Please note that the guidance is specified for specific log streams or log stream structures. The guidance is NOT specified for specific CICS regions, nor may guidance

for specific log streams or log stream structures be “overridden” on a region-by-region basis.

From a practical matter, this limitation should not pose a problem, since the log stream names normally correspond to specific CICS regions. Thus, specifying guidance for a particular log stream or log stream structure typically would cause the guidance to be associated with a CICS region.

Exhibit 3-4 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the analysis guidance variables for specific log streams. As illustrated in Exhibit 3-4, guidance for individual log streams is specified **inside** the SAS macro comment statements (/* and */). The SAS macro comment statements may not be altered, as they control CPExpert’s processing of the USOURCE(CICGUIDE) member.

The log streams are identified by the LOGNAME statement, which is used to specify the log stream name to which the specific guidance applies.

Any number of log streams may be defined with appropriate guidance specified for the log streams.

Following the LOGNAME statement are the individual guidance statements for the log stream identified. Any of the global log stream guidance variables can be specified for the log stream.

If a log stream guidance variable statement is not present or has a null value, the global default will be used. **NOTE: a null value must be indicated by a semi-colon or a SAS error will result.**

The below example shows that specific guidance for the has been specified for the SYSPLEX.OPERLOG log stream. In this example, 9999999 was specified for the guidance for the STRC2 variable. The effect of the specification is to exclude the SYSPLEX.OPERLOG from HIGHOFFLOAD analysis (as described in Rule CIC308).

```
%LET LOGGUIDE = Y;      * GUIDANCE IS PROVIDED FOR INDIVIDUAL LOG STREAMS;
/* SPECIFY GUIDANCE FOR LOG STREAMS
LOGNAME  = SYSPLEX.OPERLOG
STRC2       = 99999999    *ACCEPTABLE TIMES STRUCTURE HIGHOFFLOAD HIT         ;
*/
```


Chapter 6: Shared Temporary Storage Guidance Variables

The CICS temporary storage control facility provides the application programmer with the ability to store data in temporary storage queues, either in main storage, in auxiliary storage on a direct access storage device, or in a temporary *storage data sharing pool*. Shared temporary storage queues are stored in named pools in an MVS coupling facility. A shared TS pool consists of an XES list structure, which is accessed through a cross-memory queue server region.

Access to a shared temporary storage pool by CICS transactions running in an AOR is through a temporary storage *data sharing server* that supports the named pool. In each MVS image in the sysplex, there is one data sharing server for each pool defined in a coupling facility which can be accessed from that MVS image. All TS pool access is performed by cross-memory calls to the data sharing server for the named pool.

The shared temporary storage queue server uses a *queue index buffer pool* within its region, to read and write queue index entries. When a READQ TS or WRITE TS request completes, the queue index information is retained in the buffer. Retaining the queue index entries in the queue index buffer pool can avoid the need to reread the queue index entry if the same queue is referenced from the same MVS image before the buffer has been reused.

During server initialization, the server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. It also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. After sharing server initialization, AXM page allocation services are used to manage server region storage.

CICS provides statistics related to the shared temporary storage queue server:

- Shared temporary storage queue server statistics for the coupling *facility* are available in MXG file **CICXQ1**.
- Shared temporary storage queue server *buffer pool statistics* available in MXG file **CICXQ2**.
- Shared temporary storage queue server *storage statistics* available in MXG file **CICXQ3**.

The CICS Component analyzes data in CICXQ1, CICXQ2, and CICXQ3 to detect performance problems or potential performance problems with shared temporary storage. Exhibit 3-4 illustrates the USOURCE(CICGUIDE) variables that provide guidance to the CICS Component as it analyzes shared temporary storage. This chapter describes the guidance variables that can be used to alter CPExpert's analysis of shared temporary storage.


```
*****************************************************************   ;
*       CICS Component GUIDANCE VARIABLES                           ;
*****************************************************************   ;
      •
      •
      •
      •

* SHARED TS QUEUE SERVER GUIDANCE VARIABLES;
*;
* SERVER COUPLING FACILITY STATISTICS GUIDANCE;
%LET TSLSTFUL = 0       ; * TIMES LIST WAS FULL                          ;
%LET TSNOSPCE = 0       ; * TIMES LIST STRUCTURE WAS OUT OF SPACE        ;
%LET TSPCTELE = 70      ; * MAXIMUM PERCENT DATA ELEMENTS USED           ;
%LET TSPCTENT = 70      ; * MAXIMUM PERCENT LIST ENTRIES USED            ;
%LET TSPCTIDR = 0       ; * MINIMUM PERCENT STRUC INDEX DATA REREADS     ;
%LET TSPCTLDR = 0       ; * MINIMUM PERCENT STRUC LIST DATA REREADS      ;
%LET TSPCTNOE = 0.1     ; * PERCENT STRUCTURE ENTRY NOT FOUND            ;
%LET TSPCTTIM = 25      ; * PERCENT REQUESTS TIMEOUT AND RESTARTED       ;
%LET TSPCTVCF = 1       ; * MAXIMUM PERCENT VERSION CHECK FAILED         ;
* SERVER BUFFER POOL STATISTICS GUIDANCE;
%LET TSPCTFBP = 75      ; * PERCENT QUEUE INDEX BP BUFFERS USED          ;
%LET TSPCTLRU = 0.1     ; * PERCENT LRU ACTIVITY                         ;
%LET TSPCTUSE = 10      ; * PERCENT BUFFERS USED IN INDEX BUFFER POOL    ;
%LET TSPCTWBL = 1       ; * PERCENT WAIT ON BUFFER LOCK                  ;
%LET TSPCTWBP = 0.1     ; * PERCENT WAIT ON BUFFER POOL LOCK             ;
* SERVER STORAGE STATISTICS GUIDANCE;
%LET TSANYRQS = 0       ; * LOC=ANY STORAGE REQUEST FAILED AFTER RETRY;
%LET TSLOWRQS = 0       ; * LOC=LOW STORAGE REQUEST FAILED AFTER RETRY;
%LET TSPCTAMN = 1       ; * LOC=ANY PERCENT MINIMUM FREE STORAGE        ;
%LET TSPCTARC = 0.1     ; * LOC=ANY PERCENT REQUEST FAILED AND RETRIED;
%LET TSPCTLMN = 1       ; * LOC=LOW PERCENT MINIMUM FREE STORAGE        ;
%LET TSPCTLRC = 0.1     ; * LOC=LOW PERCENT REQUEST FAILED AND RETRIED;

*********************************************************************;
```

DEFAULT VALUES FOR SHARED TEMPORARY STORAGE ANALYSIS

EXHIBIT 3-4

Guidance variables for shared temporary storage queue server are described in three part: (1) server coupling facility statistics guidance variables, (2) server buffer pool statistics guidance variables, and (3) server storage statistics guidance variables.

Chapter 6.1: Server coupling facility statistics

Shared temporary storage queues are stored in named pools in an MVS coupling facility. A shared TS pool consists of an XES list structure, which is accessed through a cross-memory queue server region. Shared temporary storage queue server *coupling facility statistics* are available in MXG file CICXQ1.

Data items in shared temporary storage (TS) are kept in queues whose names are assigned dynamically by the program storing the data. These shared temporary storage queues are stored in *named pools* in an MVS coupling facility. These queues are related to a shared TS pool using the SYSID keyword to specify the shared queue pool to which the request is directed. Each TS pool is defined, using MVS cross-system extended services (XES), as a keyed list structure in a coupling facility.

A list structure consists of a set of lists and an optional lock table of exclusive locks (which can be used to serialize the use of lists, list entries, or other resources in the list structure). Each list is pointed to by a *list header* and can contain a number of *list entries*. With shared TS queues, the list structure is the named pool, while the lists themselves are shared TS queues within the named pool. The shared TS pool server designates the maximum number of lists (or queues) the TS pool is to have, and allocates the list structure based on parameters that are provided to the TS pool server.

A list entry consists of list entry controls and can optionally include an *adjunct area*, a *data entry*, or both. Data entries are composed of units of storage called *data elements*. Although a data entry can be composed of a number of data elements, list operations treat the data entry as a single entity; data elements cannot be read or written individually.

Chapter 6.1.1: Times list was full - TSLSTFUL variable

Three parameters control how many shared temporary storage queues can be in a specific pool and the characteristics of the queues:

- The *MAXQUEUES* parameter specifies the maximum number of data lists to be reserved when the structure is allocated.

- The *SMALLQUEUEITEMS* specifies the maximum number of items that can be stored in the small queue format in the queue index entry data area.

- The *SMALLQUEUESIZE* parameter specifies the maximum data size for a small queue including the two-byte length prefix on each data item.

If the number of queues is reached (as set by the MAXQUEUES server initialization parameter described above), any further request to establish a queue will fail, and message **DFHXQ0443** (*CF structure strname request failed, all lists are in use*) will be

issued. The failing request is given a NOSPACE indication if it originated from a CICS API request.

CPExpert uses data in CICXQ1 to determine whether a List Full condition occurred. CICXQ1 variable S1RSP6CT (List full: maximum list key reached) indicates that maximum queue size or maximum queues were reached, depending on the list.

CPExpert produces Rule CIC324 when the number of List Full conditions is greater than the value specified by the **TSLSTFUL** guidance variable in USOURCE(CICGUIDE). The default value for the **TSLSTFUL** is 0, indicating that CPExpert should produce Rule CIC324 whenever any List Full conditions occurred.

You can alter this analysis using the **TSLSTFUL** guidance variable. For example, if you wish Rule CIC324 produced only when more than one List Full condition occurred, specify:

```
%LET TSLSTFUL = 1 ; * TIMES LIST WAS FULL;
```

Chapter 6.1.2: List structure was out of space - TSNOSPCE variable

If a task tries to write to temporary storage and there is no space available, message **DFHXQ0442** (*CF structure strname request failed, structure is full*) will be issued. The failing request is given a NOSPACE indication if it originated from a CICS API request. CICS normally suspends the task (although the task can regain control in this situation by using either a HANDLE CONDITION NOSPACE command, or the RESP or NOHANDLE option on the WRITEQ TS command). If suspended, the task normally is not resumed until some other task frees the necessary space in main storage or the VSAM data set.

CPExpert uses data in CICXQ1 to determine whether a Structure Full condition occurred. CICXQ1 variable S1RSP7CT (List structure out of space) indicates that the list structure was full.

CPExpert produces Rule CIC325 when the number of Structure Full conditions is greater than the value specified by the **TSNOSPCE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSNOSPCE** is 0, indicating that CPExpert should produce Rule CIC325 whenever any Structure Full conditions occurred.

You can alter this analysis using the **TSNOSPCE** guidance variable. For example, if you wish Rule CIC324 produced only when more than one Structure Full condition occurred, specify:


```
%LET TSNOSPCE = 1 ; * TIMES LIST STRUCTURE WAS OUT OF SPACE;
```

Chapter 6.1.3: Percent data elements in use - TSPCTELE variable

With shared temporary storage, the queue server monitors the total number of data entries and data elements in use in the structure, using information returned by the coupling facility on every request. When the numbers in use exceed thresholds specified by the TS queue server *warning parameters*, a warning message (DFHXQ0411 or DFHXQ0412, for entries and elements, respectively) is issued. The default ELEMENTWARN and ENTRYWARN warning parameters have a default value of **80**, which specify that warnings and automatic ALTER actions should be first triggered when 80% of the elements or entries are used.

CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used. CPExpert uses data in CICXQ1 to calculate the maximum percent of the data elements that had been used.

CPExpert produces Rule CIC321 when the maximum percent data elements used is more than the value specified by the **TSPCTELE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTELE** is 70 indicating that CPExpert should produce Rule CIC321 whenever more than 70% of the data elements had been used.

You can alter this analysis using the **TSPCTELE** guidance variable. For example, if you wish Rule CIC321 produced only when more than 60% of the data elements had been used, specify:

```
%LET TSPCTELE = 60 ; * MAXIMUM PERCENT DATA ELEMENTS USED;
```

Chapter 6.1.4: Percent list entries in use - TSPCTENT variable

With shared temporary storage, the queue server monitors the total number of data entries and data elements in use in the structure, using information returned by the coupling facility on every request. When the numbers in use exceed thresholds specified by the TS queue server *warning parameters*, a warning message (DFHXQ0411 or DFHXQ0412, for entries and elements, respectively) is issued.

CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used. CPExpert uses data in

CICXQ1 to calculate the maximum percent of the structure list entries that had been used.

CPExpert produces Rule CIC320 when the maximum percent structure list entries used is more than the value specified by the **TSPCTENT** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTENT** is 70 indicating that CPExpert should produce Rule CIC320 whenever more than 70% of the list entries had been used.

You can alter this analysis using the **TSPCTENT** guidance variable. For example, if you wish Rule CIC320 produced only when more than 60% of the list entries had been used, specify:

```
%LET TSPCTENT = 60 ; * MAXIMUM PERCENT LIST ENTRIES USED;
```

Chapter 6.1.5: Percent repeated index data reads - TSPCTIDR variable

The READQ TS command reads data from a temporary storage queue. The command optionally specifies INTO(data-area) to describe the data area into which the data is to be written. If the length of the data exceeds the value specified, the data is truncated to that value. A program error (LENGERR) occurs when the length of the stored data is greater than the value specified by the LENGTH option (this condition only applies to the INTO option).

If the request ended prematurely because the buffer was too small to hold the first entry to be read (for instance, the buffer is 4096 bytes but the data entry information is 65536 bytes), the application must determine the size of the data entry for the list entry that caused the failure, and re-issue the READQ TS command with larger buffer areas. This error handling and command re-issue creates unnecessary overhead and delays response.

CPExpert uses data in CICXQ1 to calculate the percent of *index data* reads which had to be repeated because the data was larger than the default data transfer size.

CPExpert produces Rule CIC327 when the percent of queue index reads had to be repeated because the data was larger than the default data transfer size is greater than the value specified by the **TSPCTIDR** guidance variable. The default value for the **TSPCTIDR** is 0, indicating that CPExpert should produce Rule CIC327 when any queue index reads must be repeated because the data was larger than the default data transfer size.

You can alter this analysis using the **TSPCTIDR** guidance variable. For example, if you wish Rule CIC327 produced only when more than 10% of the queue index reads must be repeated, specify:

```
%LET TSPCTIDR = 10 ; * MINIMUM PERCENT STRUCTURE INDEX DATA REREADS;
```

Chapter 6.1.6: Percent repeated list data reads - TSPCTLDR variable

The READQ TS command reads data from a temporary storage queue. The command optionally specifies INTO(data-area) to describe the data area into which the data is to be written. If the length of the data exceeds the value specified, the data is truncated to that value. A program error (LENGERR) occurs when the length of the stored data is greater than the value specified by the LENGTH option (this condition only applies to the INTO option).

If the request ended prematurely because the buffer was too small to hold the first entry to be read (for instance, the buffer is 4096 bytes but the data entry information is 65536 bytes), the application must determine the size of the data entry for the list entry that caused the failure, and re-issue the READQ TS command with larger buffer areas. This error handling and command re-issue creates unnecessary overhead and delays response.

If the request ended prematurely because the buffer was too small to hold the first entry to be read (for instance, the buffer is 4096 bytes but the data entry information is 65536 bytes), the application must determine the size of the data entry for the list entry that caused the failure, and re-issue the READQ TS command with larger buffer areas. This error handling and command re-issue creates unnecessary overhead and delays response.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to calculate the percent of list data reads which had to be repeated because the data was larger than the default data transfer size.

CPExpert produces Rule CIC326 when the percent of list data reads which had to be repeated because the data was larger than the default data transfer size is greater than the value specified by the **TSPCTLDR** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLDR** is 0 indicating that CPExpert should produce Rule CIC326 when any list data reads must be repeated because the data was larger than the default data transfer size.

You can alter this analysis using the **TSPCTLDR** guidance variable. For example, if you wish Rule CIC326 produced only when more than 10% of the list data reads must be repeated, specify:

```
%LET TSPCTLDR = 10 ; * MINIMUM PERCENT STRUCTURE LIST DATA REREADS;
```

Chapter 6.1.7: Percent structure entry not found - TSPCTNOE variable

The CICS temporary storage control facility provides the application programmer with the ability to store data in temporary storage queues, either in main storage, in auxiliary storage on a direct access storage device, or in a temporary storage data sharing pool. Several commands can be used to reference, modify, or delete data in a temporary storage queue or to delete the entire queue.

When the commands are executed, the response from the command can be “normal” in that no further action is required, or the response can be “abnormal” in that either the task is terminated or action is required. Two abnormal conditions that apply to shared temporary storage are reported in the CICS statistics: *ITEMERR* and *QIDERR*.

- The ITEMERR abnormal condition can occur with the READQ TS command when (1) the item number specified is invalid (that is, the item number is outside the range of item numbers written to the queue), or (2) an attempt is made to read beyond the end of the queue using the NEXT (default) option.

- The QIDERR abnormal condition can occur with the READQ TS command, with the WRITEQ TS command, and with the DELETEQ TS command. The QIDERR abnormal condition occurs when the specified queue cannot be found.

CPExpert uses data in CICXQ1 to calculate the percent of requests that encountered a “specified entry (queue or item) was not found” condition.

CPExpert produces Rule CIC323 when the percent requests that encountered a “specified entry (queue or item) was not found” condition is greater than the value specified by the **TSPCTNOE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTNOE** is 0.1 indicating that CPExpert should produce Rule CIC323 whenever more than one tenth percent of the requests encountered a “specified entry (queue or item) was not found” condition.

You can alter this analysis using the **TSPCTNOE** guidance variable. For example, if you wish Rule CIC323 produced only when more than 10% of the requests encountered a “specified entry (queue or item) was not found” condition, specify:


```
%LET TSPCTNOE = 10 ; * PERCENT REQUESTS STRUCTURE ENTRY NOT FOUND;
```

Chapter 6.1.8: Percent requests timeout - TSPCTTIM variable

Some list structure commands can complete prematurely, because the request exceeded the coupling facility model-dependent time-out criteria. When list structure commands complete prematurely, the application typically restarts the command (using information has been returned in the answer area). Commands that complete prematurely cause unnecessary overhead in the coupling facility and in the application.

CPExpert uses data in CICXQ1 to calculate the percent of requests that timed out. CPExpert produces Rule CIC322 when the percent requests that timed out is greater than the value specified by the **TSPCTTIM** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTTIM** is 1 indicating that CPExpert should produce Rule CIC322 whenever more than one percent of the requests timed out because the request exceeded the coupling facility model-dependent time-out criteria.

You can alter this analysis using the **TSPCTTIM** guidance variable. For example, if you wish Rule CIC322 produced only when more than 10% of the requests timed out, specify:

```
%LET TSPCTTIM = 10 ; * PERCENT REQUESTS TIMEOUT AND RESTARTED;
```

Chapter 6.1.9: Percent version check failed - TSPCTVCF variable

Temporary storage control commands WRITEQ TS and DELETEQ TS invoke implicit enqueuing. However, CICS enqueuing is not invoked for READQ TS commands. This makes possible for one task to read a temporary storage queue record while another is updating the same record.

After issuing the READQ TS command, if the application wishes to modify the information and then issue a WRITEQ TS command, it is possible that the temporary storage queue record would have been updated by another task. In this case, the WRITEQ TS command would fail because of a version check.

CPExpert uses data in CICXQ1 to calculate the percent of requests that encountered a “version check failed for an entry being updated” condition.

CPExpert produces Rule CIC328 when the percent requests that encountered a "version check failed for an entry being updated" condition is greater than the value specified by the **TSPCTVCF** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTVCF** is 0.1 indicating that CPExpert should produce Rule CIC328 whenever more than one tenth percent of the requests encountered a "specified entry (queue or item) was not found" condition.

You can alter this analysis using the **TSPCTVCF** guidance variable. For example, if you wish Rule CIC323 produced only when more than 10% of the requests encountered a "version check failed for an entry being updated" condition, specify:

```
%LET TSPCTVCF = 10 ; * PERCENT VERSION CHECK FAILED CONDITION;
```

Chapter 6.2: Server buffer pool statistics

The shared temporary storage queue server uses a *queue index buffer pool* within its region, to read and write queue index entries. When a READQ TS or WRITEQ TS request completes, the queue index information is retained in the buffer. Retaining the queue index entries in the queue index buffer pool can avoid the need to reread the queue index entry if the same queue is referenced from the same MVS image before the buffer has been reused.

When a request for the same queue arrives, the shared TS queue server determines whether the queue index information is in the buffer. If the information is in the buffer, a coupling facility access is avoided. When the request completes, the shared TS queue server places the information into a buffer, onto a least recently used (LRU) chain. If all other buffers are in use, a request for a new buffer will discard the contents of the least recently used buffer and reuse the storage as a free buffer.

Shared temporary storage queue server *buffer statistics* are available in MXG file CICXQ2.

Chapter 6.2.1: Percent index buffers in use - TSPCTFBP variable

If a large percent of buffers were in use, it is possible that (1) an inadequate number of buffers were defined or (2) the application is not freeing buffers in a timely manner.

CPExpert uses data in CICXQ2 to calculate the percent of queue index buffer pool buffers that were used.

CPExpert produces Rule CIC330 when the percent queue index buffer pool buffers that were used is greater than the value specified by the **TSPCTFBP** guidance variable in

USOURCE(CICGUIDE). The default value for the **TSPCTFBP** is 75, indicating that CPExpert should produce Rule CIC330 whenever more than 75% of the queue index buffer pool buffers were used.

This finding is produced as an “early warning” of a potential problem. A large percent of LRU activity would be reported by **Rule CIC331**, and could imply “thrashing” in the queue index buffer pool.

You can alter this analysis using the **TSPCTFBP** guidance variable. For example, if you wish Rule CIC330 produced only when more than 90% of the queue index buffer pool buffers were used, specify:

```
%LET TSPCTFBP = 90 ; * PERCENT MINIMUM FREE BUFFERS IN BUFFER POOL;
```

Chapter 6.2.2: Percent LRU activity - TSPCTLRU variable

If a buffer is required from the LRU chain, this means that all other buffers were in use. If all other buffers were in use, it is possible that (1) an inadequate number of buffers were defined or (2) the application is not freeing buffers in a timely manner. In either case, a large percent of LRU activity could imply “thrashing” in the queue index buffer pool.

CPExpert uses data in CICXQ2 to calculate the percent LRU activity. CPExpert produces Rule CIC331 when the percent LRU activity is greater than the value specified by the **TSPCTLRU** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLRU** is 0.1, indicating that CPExpert should produce Rule CIC331 whenever the percent LRU activity is greater than 0.1% of the queue index buffer pool requests. This low percent LRU activity was set as a default to alert you to a potential problem that requires attention.

You can alter this analysis using the **TSPCTLRU** guidance variable. For example, if you wish Rule CIC331 produced only when the percent LRU activity is greater than 20% of the queue index buffer pool requests, specify:

```
%LET TSPCTLRU = 20 ; * PERCENT LRU ACTIVITY;
```


Chapter 6.2.3: Percent buffer pool buffers used- TSPCTUSE variable

The number of buffers in the queue index buffer pool is defined using the BUFFERS= keyword in the TS queue server parameters. The default specification is BUFFERS={100}, which specifies that 100 buffers should be allocated to for the server address space. IBM states that it is not worth defining extra buffers beyond the point where the definition might cause MVS paging, as it is more efficient to reread the index entry than to page in the buffer from auxiliary storage.

CPExpert uses data in CICXQ2 to calculate the maximum percent of the queue index buffer pool buffers that had been used.

CPExpert produces Rule CIC332 when the percent queue index buffer pool buffers used is less than the value specified by the **TSPCTUSE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTUSE** is 10, indicating that CPExpert should produce Rule CIC332 whenever less than 10% of the queue index buffer pool buffers were used. CPExpert suppresses this finding if 100 buffers or less are defined.

You can alter this analysis using the **TSPCTUSE** guidance variable. For example, if you wish Rule CIC331 produced only when less than 5% of the queue index buffer pool buffers were used, specify:

```
%LET TSPCTUSE = 5 ; * PERCENT BUFFERS USED IN INDEX BUFFER POOL;
```

Chapter 6.2.4: Percent wait on buffer lock - TSPCTWBL

Temporary storage control commands WRITEQ TS and DELETEQ TS invoke *implicit enqueuing*. However, CICS enqueuing is not invoked for READQ TS commands. This makes possible for one task to read a temporary storage queue record while another is updating the same record.

CICS provides two explicit enqueuing commands (EXEC CICS ENQ RESOURCE and EXEC CICS DEQ RESOURCE). These commands can be used to protect a temporary storage queue from being read and updated concurrently.

After a task has issued an ENQ RESOURCE(data-area) command, any other task that issues an ENQ RESOURCE command with the same data-area parameter is suspended until the task issues a matching DEQ RESOURCE(data-area) command, or until the UOW ends.

CPExpert uses data in CICXQ2 to calculate the percent of GET waits on buffer lock.

CPExpert produces Rule CIC334 when the percent GET waits on buffer lock is greater than the value specified by the **TSPCTWBL** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTWBL** is 1, indicating that CPExpert should produce Rule CIC334 whenever more than one percent of the GET requests waited for a buffer lock. CPExpert suppresses this finding unless there are more than 1,000 GET requests in the statistics interval.

You can alter this analysis using the **TSPCTWBL** guidance variable. For example, if you wish Rule CIC331 produced only when more than 5% of the GET requests waited for a buffer lock, specify:

```
%LET TSPCTWBL = 5 ; * PERCENT WAIT ON BUFFER LOCK;
```

Chapter 6.2.5: Percent wait on buffer pool lock - TSPCTWBP

Shared temporary storage queues are stored in named pools in an MVS coupling facility. A shared TS pool consists of an XES list structure, which is accessed through a cross-memory queue server region. There are two situations in which shared temporary storage requests can wait on the shared TS pool:

- **The queue pool is locked for exclusive use**. Message **DFHXQ0407** (CF structure strname is not available for shared use) issued if a request for a TS pool cannot be satisfied because the queue pool is locked for exclusive use by some other job (such as a queue pool unload or reload job).

  The server is terminated in this situation. Consequently, this situation is unlikely to be the cause of frequent waits on a shared temporary storage buffer pool lock.

- **SUSPEND on resource type TSPOOL**. Resource type TSPOOL indicates that the maximum number of concurrent requests (10) for a temporary storage pool in the coupling facility has been reached. The task resumes when one of the requests completes.

CPExpert uses data in CICXQ2 to calculate the percent of requests to the shared temporary storage buffer pool that waited on a buffer pool lock.

CPExpert produces Rule CIC333 when the percent of requests to the shared temporary storage buffer pool that waited on a buffer pool lock is greater than the value specified by the **TSPCTWPB** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTWPB** is 0.1, indicating that CPExpert should produce Rule CIC333 whenever more than one tenth of the requests waited on a buffer pool lock.

You can alter this analysis using the **TSPCTWPB** guidance variable. For example, if you wish Rule CIC333 produced only when more than five percent of the requests waited on a buffer pool lock, specify:

```
%LET TSPCTWPB = 5 ; * PERCENT WAIT ON BUFFER POOL LOCK;
```

Chapter 6.3: Pool server storage statistics

Shared temporary storage queues are stored in named pools in an MVS coupling facility. Access to a shared temporary storage pool by CICS transactions running in an AOR is through a TS *data sharing server* that supports a specific named pool. All shared TS pool access is performed by cross-memory calls to the data sharing server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. It also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage, for example sequential file read and write routines.

During server initialization, the data sharing server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as the *AXMPGANY pool*. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as the *AXMPGLOW* pool.

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. If there is still insufficient storage to satisfy the request, the request fails.

Shared temporary storage queue server *storage statistics* are available in MXG file CICXQ3.

Chapter 6.3.1: AXMPGANY requests failed after retry - TSANYRQS

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

CPExpert produces Rule CIC341 when the number of storage requests in the AXMPGANY pool that failed after retry (variable S3ANYRQS[16] in CICXQ3) is greater than the value specified by the **TSANYRQS** guidance variable in USOURCE(CICGUIDE). The default value for the **TSANYRQS** is 0, indicating that CPExpert should produce Rule CIC341 when any storage request in the AXMPGANY pool failed after retry.

You can alter this analysis using the **TSANYRQS** guidance variable. Since Rule CIC341 describes a situation that has a high impact on the performance of the CICS region, **you should not normally change the TSANYRQS guidance variable**. However, if you have very unusual circumstance[17] and you wish Rule CIC341 produced only when more than five storage requests in the AXMPGANY pool failed after retry, specify:

```
%LET TSANYRQS = 5 ; * LOC=ANY REQUESTS FAILED AFTER RETRY;
```

Chapter 6.3.2: AXMPGLOW requests failed after retry - TSLOWRQS

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

CPExpert produces Rule CIC344 when the number of storage requests in the AXMPGLOW pool that failed after retry (variable S3LOWRQS[18] in CICXQ3) is greater than the value specified by the **TSLOWRQS** guidance variable in USOURCE(CICGUIDE).

[16] Please note that the *CICS Performance Guides* describe S3ANYRQF as failed requests and S3ANYRQS as requests to release storage. These descriptions are "reversed" and IBM Hursley has agreed to correct the *CICS Performance Guides*. S3ANYRQF is the requests to "free" or release storage and S3ANYRQS is number of failed requests because of "short on storage".

[17] I would greatly appreciate being advised of any such circumstance, so I can alter CPExpert's logic and documentation accordingly. Please send a note to Don_Deese@cpexpert.com. Thanks!

[18] Please note that the *CICS Performance Guides* describe S3LOWRQF as failed requests and S3LOWRQS as requests to release storage. These descriptions are "reversed" and IBM Hursley has agreed to correct the *CICS Performance Guides*. S3LOWRQF is the requests to "free" or release storage and S3LOWRQS is number of failed requests because of "short on storage".

The default value for the **TSLOWRQS** is 0, indicating that CPExpert should produce Rule CIC344 when any storage request in the AXMPGLOW pool failed after retry.

You can alter this analysis using the **TSLOWRQS** guidance variable. Since Rule CIC344 describes a situation that has a high impact on the performance of the CICS region, **you should not normally change the TSLOWRQS guidance variable**. However, if you have very unusual circumstance and you wish Rule CIC341 produced only when more than five storage requests in the AXMPGLOW pool failed after retry, specify:

```
%LET TSLOWRQS = 5 ; * LOC=BELOW REQUESTS FAILED AFTER RETRY;
```

Chapter 6.3.3: Low percent AXMPGANY free storage - TSPCTAMN

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

CPExpert uses data in CICXQ3 to calculate the minimum percent of free storage in the AXMPGANY pool.

CPExpert produces Rule CIC342 when the percent free storage in the AXMPGANY pool is less than the value specified by the **TSPCTAMN** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTAMN** is 25, indicating that CPExpert should produce Rule CIC342 whenever less than 25% of storage in the AXMPGANY pool is free.

You can alter this analysis using the **TSPCTAMN** guidance variable. For example, if you wish Rule CIC342 produced only when less than 25% of storage in the AXMPGANY pool is free , specify:

```
%LET TSPCTAMN = 5 ; * LOC=ANY PERCENT MINIMUM FREE STORAGE;
```

Chapter 6.3.4: Percent AXMPGANY requests retried - TSPCTARC

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a

large enough area has been created. This action appears as a compress attempt in the statistics.

When requests in the AXMPGANY pool are retried, this means that all free storage in the AXMPGANY pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the "compress attempt" to reduce fragmentation in the AXMPGANY storage POOL will correspondingly increase.

CPExpert uses data in CICXQ3 to calculate the percent of storage requests in the AXMPGANY pool initially failed and were retried.

CPExpert produces Rule CIC340 when the percent of storage requests in the AXMPGANY pool initially failed and were retried is greater than the value specified by the **TSPCTARC** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTARC** is .1, indicating that CPExpert should produce Rule CIC340 whenever more than one tenth percent of the storage requests in the AXMPGANY pool initially failed and were retried.

You can alter this analysis using the **TSPCTARC** guidance variable. For example, if you wish Rule CIC340 produced only when more than five percent of storage requests in the AXMPGANY pool initially failed and were retried , specify:

```
%LET TSPCTARC = 5 ; * PERCENT LOC=ANY REQUEST FAILED AND RETRIED;
```

Chapter 6.3.5: Low percent AXMPGLOW free storage - TSPCTLMN

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

CPExpert uses data in CICXQ3 to calculate the minimum percent of free storage in the AXMPGLOW pool.

CPExpert produces Rule CIC345 when the percent free storage in the AXMPGLOW pool is less than the value specified by the **TSPCTLMN** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLMN** is 25, indicating that CPExpert should produce Rule CIC345 whenever less than 25% of storage in the AXMPGLOW pool is free.

You can alter this analysis using the **TSPCTLMN** guidance variable. For example, if you wish Rule CIC345 produced only when less than 25% of storage in the AXMPGLOW pool is free , specify:

```
%LET TSPCTLMN = 5 ; * LOC=LOW PERCENT MINIMUM FREE STORAGE;
```

Chapter 6.3.6: Percent AXMPGLOW requests retried - TSPCTLRC

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics.

When requests in the AXMPGLOW pool are retried, this means that all free storage in the AXMPGLOW pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the "compress attempt" to reduce fragmentation in the AXMPGLOW storage POOL will correspondingly increase.

CPExpert uses data in CICXQ3 to calculate the percent of storage requests in the AXMPGLOW pool initially failed and were retried.

CPExpert produces Rule CIC340 when the percent of storage requests in the AXMPGLOW pool initially failed and were retried is greater than the value specified by the **TSPCTLRC** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLRC** is .1, indicating that CPExpert should produce Rule CIC340 whenever more than one tenth percent of the storage requests in the AXMPGLOW pool initially failed and were retried.

You can alter this analysis using the **TSPCTLRC** guidance variable. For example, if you wish Rule CIC340 produced only when more than five percent of storage requests in the AXMPGLOW pool initially failed and were retried , specify:

```
%LET TSPCTLRC = 5 ; * PERCENT LOC=LOW REQUEST FAILED AND RETRIED;
```


Chapter 6.4 Specifying guidance for specific shared TS pools

The guidance variables for shared temporary storage pools are globally applied during CPExpert's analysis of shared temporary storage performance from the perspective of CICS. These global guidance variables might not be applicable to some shared temporary storage pools, however. Guidance can **optionally** be applied to specific shared temporary storage pools or to the coupling facility structures used by the shared temporary storage pools.

Guidance for specific shared temporary storage pools is accomplished by specifying **%LET TSGUIDE=Y;** in USOURCE(CICGUIDE), identifying the shared temporary storage pools to which the guidance applies, and specifying guidance variables for the shared temporary storage pools.

Please note that the guidance is specified for specific shared temporary storage pools. The guidance is NOT specified for specific CICS regions, nor may guidance for shared temporary storage pools be "overridden" on a region-by-region basis. This is because shared temporary storage pools are not associated with a specific CICS region.

Exhibit 3-4 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the analysis guidance variables for specific shared temporary storage pools. As illustrated in Exhibit 3-4, guidance for individual shared temporary storage pools is specified **inside** the SAS macro comment statements (/* and */). The SAS macro comment statements may not be altered, as they control CPExpert's processing of the USOURCE(CICGUIDE) member.

The shared temporary storage pools are identified by the POOL statement, which is used to specify the shared temporary storage pool to which the specific guidance applies.

Any number of shared temporary storage pools may be defined with appropriate guidance specified for each pool.

Following the POOL statement are the individual guidance statements for the shared temporary storage pool identified. Any of the global shared temporary storage pool guidance variables can be specified for the shared temporary storage pool.

If a particular shared temporary storage pool guidance variable statement is not present or has a null value, the global default will be used. **NOTE: a null value must be indicated by a semi-colon or a SAS error will result.**

The below example shows that specific guidance for the has been specified for the G1POOL2 shared temporary storage pool server. In this example, 10 was specified as the guidance for the TSPCTLRU variable.


```
%LET TSGUIDE = Y;        * GUIDANCE IS PROVIDED FOR INDIVIDUAL SHARED TS POOLS;
/* SPECIFY GUIDANCE FOR SHARED TEMPORARY STORAGE POOLS
POOL  = G1POOL2
TSPCTLRU  =10  * PERCENT LRU ACTIVITY         ;
*/
```


Chapter 7: Shared Data Tables and CFDT Guidance Variables

For VSAM data sets, CICS file control provides applications with *file control commands* that read, update, add, and browse the data sets. There are many features provided by IBM that allow applications to use file control commands to reference VSAM data sets, and allow separate applications to *share* the data. Two such features are the *shared data table* approach and the Coupling Facility Data Table approach.

- **Shared data tables**. An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

  With shared data table support, the KSDS file is called the *source data set*. The copy of the file in memory is called the *data table*. The process of copying the records from the file to the data table is called *loading the data table*. Whenever a CICS application wishes to reference the VSAM file using normal file control commands, CICS attempts to use the representation of the file in the data table, rather than accessing the source data.

- **Coupling Facility Data Tables.** CICS Coupling Facility Data Tables provide a significant enhancement to shared data tables in a parallel sysplex. The CFDT design provides an excellent way to share file data using CICS file control, without resorting to VSAM record level sharing (RLS). The CFDT design eliminates the requirement for having a File Owning Region (as is required with normal shared data tables).

  CICS CFDT support is designed to provide sharing of working data within a sysplex, while maintaining update integrity of the data. The working data is held in a coupling facility data table, which is contained in a *named pool* located in coupling facility list structure. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a list structure in a coupling facility.

The CICS Component anlayzes data in CICCFS6D, CICCFS7D, CICCFS8D, and CICCFS9D to detect performance problems or potential performance problems with coupling facility data tables.

Exhibit 3-4 illustrates the USOURCE(CICGUIDE) variables that provide guidance to the CICS Component as it analyzes shared data tables and coupling facility data tables. This chapter describes the guidance variables that can be used to alter CPExpert's analysis of shared data tables or coupling facility data tables.


```
************************************************************   ;
*      CICS Component GUIDANCE VARIABLES                                ;
************************************************************   ;
       •
       •
       •
       •

* SHARED DATA TABLE/COUPLING FACILITY DATA TABLE GUIDANCE VARIABLES ;
*                                                                  ;
%LET CICSRNF  = 100  ;  * ACCEPTABLE RECORDS NOT FOUND, CICS-SDT    ;
%LET MINSDTIO = 500  ;  * MINIMUM VSAM DATA TABLE IO (RULE 405/406) ;
%LET PCTDTSRC = 25   ;  * PCT SDT ACCESSES TO SOURCE DATA SET       ;
%LET SDTFULL  = 1    ;  * NUMBER SHARED DATA TABLE FULL CONDITIONS  ;
%LET UMTRNF   = 0    ;  * ACCEPTABLE RECORDS NOT FOUND, UMT-SDT     ;
* CFDT SERVER COUPLING FACILITY STATISTICS GUIDANCE;
%LET CFLSTFUL = 0    ;  * TIMES MAXNUMRECS WAS REACHED FOR CFDT     ;
%LET CFNOSPCE = 0    ;  * TIMES LIST STRUCTURE WAS OUT OF SPACE     ;
%LET CFPCTELE = 70   ;  * MAXIMUM PERCENT DATA ELEMENTS USED        ;
%LET CFPCTENT = 70   ;  * MAXIMUM PERCENT LIST ENTRIES USED         ;
%LET CFPCTRNF = .1   ;  * ACCEPTABLE PCT RECORDS NOT FOUND, CFDT    ;
* CFDT POOL SERVER STORAGE STATISTICS GUIDANCE;
%LET CFANYRQS = 0    ;  * LOC=ANY STORAGE REQUEST FAILED AFTER RETRY;
%LET CFLOWRQS = 0    ;  * LOC=LOW STORAGE REQUEST FAILED AFTER RETRY;
%LET CFPCTAMN = 25   ;  * LOC=ANY PERCENT MINIMUM FREE STORAGE      ;
%LET CFPCTARC = .1   ;  * LOC=ANY STORAGE REQUEST FAILED AND RETRIED;
%LET CFPCTLMN = 25   ;  * LOC=LOW PERCENT MINIMUM FREE STORAGE      ;
%LET CFPCTLRC = .1   ;  * LOC=LOW STORAGE REQUEST FAILED AND RETRIED;
*****************************************************************;
```

DEFAULT VALUES FOR SHARED TEMPORARY STORAGE ANALYSIS

EXHIBIT 3-4

Guidance variables for shared data tables and coupling facility data tables are described in three parts: (1) shared data table statistics guidance variables, (2) CFDT pool server coupling facility statistics guidance variables, and (3) CFDT pool server storage statistics guidance variables.

Chapter 7.1: Shared data table statistics guidance variables

The CICS Component analyzes data in CICFCR to detect performance problems or potential performance problems with shared data tables. The following paragraphs describe the guidance variables that apply to shared data tables.

Chapter 7.1.1: Records not found in CICS-maintained shared data table - CICSRNF

CICS supports two types of data table: (1) CICS-maintained data tables and (2) user-maintained data tables. A CICS-maintained data table is one that CICS keeps in synchronization with their source data sets. That is, any update or delete action on a record in the data table is automatically applied to the source data set *before* being applied to the data table. A user-maintained data table (UMT) is one that is not maintained by CICS, but is completely maintained by user code. A UMT is detached from its source data set after the table is loaded from the source data set, and changes made to the UMT are *not* reflected in the VSAM source data set.

Applications can reference the data table during initial loading. These references will produce a "record not found" condition if the references are to records outside the range of those already loaded into the data table. For CICS-maintained data tables, small numbers of the "record not found" condition after initial loading should not normally be a cause for alarm. However, a large number of "records not found" will cause unnecessary overhead and delay to applications.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses the A17DTRNF variable to assess the number of times records were not found for CICS-maintained data tables. The A17DTRNF variable contains a count of the number of times CICS attempted to read a record but was unable to satisfy the read request because the record was not in the data table; CICS was required to retrieve the record from the source data set.

CPExpert produces Rule CIC402 when the A17DTRNF value is greater than the **CICSRNF** guidance variable in USOURCE(CICGUIDE). The default value for the **CICSRNF** guidance variable is 100, indicating that CPExpert should produce Rule CIC402 when more than one hundred read requests resulted in a "record not found" condition.

You can change the **CICSRNF** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC402 is produced too often. For example, if you wish Rule CIC402 produced only when CICS encountered a "record not found" condition more than 500 times, specify:

```
%LET CICSRNF = 500 ; * ACCEPTABLE RECORDS NOT FOUND;
```


Chapter 7.1.2: Minimum shared data table I/O - MINSDTIO

An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

Since a major benefit of a data table is that records in the data table can be accessed quickly, this benefit is available only if the records actually are accessed in the data table rather than in the VSAM source data set. Any operation requiring access to the source data set reduces the efficiency of the data table.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed the VSAM source data for a CICS-maintained shared data table. Based on this calculated percent, CPExpert will produce Rule CIC405 (high data set activity for CICS-maintained shared data table) or Rule CIC406 (VSAM data set might not be good candidate or shared data table).

Before producing Rule CIC405 or Rule CIC406, however, CPExpert applies the **MINSDTIO** variable to ensure that findings are produced only for shared data tables with a reasonably high amount to of I/O activity. The default value for the **MINSDTIO** is 500, indicating that Rule CIC405 and Rule CIC406 will be suppressed unless at least 500 VSAM file control commands are issued to the shared data table.

You can change the **MINSDTIO** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC405 or Rule CIC406 are produced too often. For example, if you wish this analysis to be applied to shared data tables only when more than 1000 file control requests were to a shared data table, specify:

```
%LET MINSDTIO = 1000 ; * MINIMUM VSAM I/O ACTIVITY (CIC405/CIC406)
```

Chapter 7.1.3: Percent data table access to source data - PCTDTSRC

An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

Since a major benefit of a data table is that records in the data table can be accessed quickly, this benefit is available only if the records actually are accessed in the data table

rather than in the VSAM source data set. Any operation requiring access to the source data set reduces the efficiency of the data table.

If a large percent of file control commands access the VSAM source data set, the benefits of using a data table can outweigh the overhead and virtual storage costs of maintaining a data table.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed the VSAM source data for a CICS-maintained shared data table.

CPExpert produces Rule CIC405 when the percent VSAM source data set accesses is more than the value specified by the **PCTDTSRC** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTDTSRC** is 50 indicating that CPExpert should produce Rule CIC405 whenever more than 50% of the file accesses required that CICS access the VSAM source data set.

You can change the **PCTDTSRC** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC405 is produced too often. For example, if you wish this analysis to be applied to shared data tables only when more than 75% of the file control requests to a shared data table went to the VSAM source data set, specify:

```
%LET PCTDTSRC = 75% ; * PERCENT SDT REQUESTS TO VSAM SOURCE DATA SET
```

Chapter 7.1.4: Number of times shared data table was full - SDTFULL

Records are placed into a shared data table in one of three ways: (1) records placed in the data table by the initial loading of the data table from the source data set, (2) records subsequently added to the data table from the source data set, and (3) new records written to the data table after the data table has been loaded.

Regardless of how records are added to a shared data table, the number of records in the table cannot exceed the number specified by the MAXNUMRECS variable specified for the table. If that limit is reached during loading or while adding to the table, a “table full” condition applies. Encountering a “table full” condition can have several undesirable effects.

CPExpert uses the A17DTATF variable in the MXG CICFCR file to assess the number of records that CICS attempted to add to the table but was unable to do so because the table was full. The A17DTATF value means that the data table already contained the maximum number of records specified in the MAXNUMRECS parameter of the DEFINE FILE command.

CPExpert produces Rule CIC401 when the A17DTATF value is greater than the **SDTFULL** guidance variable in USOURCE(CICGUIDE). The default value for the **SDTFULL** guidance variable is one, indicating that CPExpert should produce Rule CIC401 when more than one shared data table full condition was detected. Note that the initial loading of the shared data table normally will produce a count of one for the A17DTATF, so the default value of the SDTFULL variable accounts for this normal situation.

You can change the **STDFULL** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC401 is produced too often. For example, if you wish Rule CIC401 produced only when a shared data table was full more than 100 times, specify:

```
%LET SDTFULL = 100 ; * TIMES SHARED DATA TABLE WAS FULL;
```

Chapter 7.1.5: Records not found in user-maintained shared data table - UMTRNF

CICS supports two types of data table: (1) CICS-maintained data tables and (2) user-maintained data tables. A CICS-maintained data table is one that CICS keeps in synchronization with their source data sets. That is, any update or delete action on a record in the data table is automatically applied to the source data set *before* being applied to the data table. A user-maintained data table (UMT) is one that is not maintained by CICS, but is completely maintained by user code. A UMT is detached from its source data set after the table is loaded from the source data set, and changes made to the UMT are *not* reflected in the VSAM source data set.

Applications can reference the data table during initial loading. These references will produce a “record not found” condition if the references are to records outside the range of those already loaded into the data table. For user-maintained data tables, the “record not found” condition should not occur after loading of the data table. Since the VSAM source data set is not available after the initial loading of the data table, any “record not found” condition for user-maintained data tables normally indicates a coding or logic error with the application.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses the A17DTRNF variable to assess the number of times records were not found for usesr-maintained data tables. The A17DTRNF variable contains a count of the number of times CICS attempted to read a record but was unable to satisfy the read request because the record was not in the data tablet.

CPExpert produces Rule CIC403 when the A17DTRNF value is greater than the **UMTRNF** guidance variable in USOURCE(CICGUIDE). The default value for the **UMTRNF** guidance

variable is 0, indicating that CPExpert should produce Rule CIC403 when any read request resulted in a "record not found" condition.

You can change the **UMTRNF** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC403 is produced too often. For example, if you wish Rule CIC403 produced only when CICS encountered a "record not found" condition more than 100 times for a user-maintained data table, specify:

```
%LET UMTRNF = 100 ; * ACCEPTABLE RECORDS NOT FOUND;
```

Chapter 7.2: CFDT pool server coupling facility structure statistics

A coupling facility data table is contained in a *named CFDT pool* located in coupling facility list structure. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a list structure in a coupling facility. CFDT pool server *coupling facility statistics* are available in MXG file CICCFS6D.

A list structure consists of a set of lists and an optional lock table of exclusive locks (which can be used to serialize the use of lists, list entries, or other resources in the list structure). Each list is pointed to by a *list header* and can contain a number of *list entries*. With coupling facility data tables, the list structure is the named CFDT pool, while the lists themselves are CFDTs within the named pool. The CFDT pool server designates the maximum number of CFDTs the CFDT pool is to have, and allocates the list structure based on parameters that are provided to the CFDT pool server when the pool server region is started.

A list entry consists of list entry controls and can optionally include an *adjunct area*, a *data entry*, or both. Data entries are composed of units of storage called *data elements*. Although a data entry can be composed of a number of data elements, list operations treat the data entry as a single entity; data elements cannot be read or written individually.

Chapter 7.2.1: Times CFDT was full - CFLSTFUL variable

The MAXNUMRECS parameter provided with the VSAM file definition specifies the maximum number of records that can be in a coupling facility data table. The default value for the MAXNUMRECS parameter is NOLIMIT, indicating that there is no limit on the maximum number of records. When the number of records in the CFDT reaches the MAXNUMRECS value, the CFDT is marked "full" and no further records can be added to

the CFDT until records are deleted to make space. The WRITE request is rejected, and the transaction must handle the NOSPACE return with an exception handling routine.

CFDT list structure statistics for the coupling facility are available in MXG file CICCFS6D. The S6RSP6CT variable contains a count of the number of times that a list (or table) became full. CPExpert calculates the percent of requests that encountered a “list full” condition.

CPExpert produces Rule CIC424 when the number of List Full conditions is greater than the value specified by the **CFPCTFUL** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTFUL** is 0, indicating that CPExpert should produce Rule CIC424 whenever any List Full conditions occurred.

You can alter this analysis using the **CFLSTFUL** guidance variable. For example, if you wish Rule CIC424 produced only when more than one List Full condition occurred, specify:

```
%LET CFLSTFUL = 1 ; * TIMES LIST WAS FULL;
```

Chapter 7.1.2: List structure was out of space - CFNOSPCE variable

Coupling Facility Data Tables are kept in a *named pool* in an MVS coupling facility. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a *list structure* in a coupling facility.

If the CFDT list structure is allowed to become completely full,message **DFHCF0442** *(CF structure strname request failed, structure is full*) is issued and CICS rejects any attempt to add new records to a CFDT or to create new tables in the pool. Additionally, completely filling a CFDT list structure can have a significant impact on performance and application function. IBM

CFDT list structure statistics for the coupling facility are available in MXG file CICCFS6D. CPExpert uses data in CICCFS6D to determine whether a Structure Full condition occurred for a CFDT pool server list structure. CICCFS6D variable S6RSP7CT (List structure became full) indicates that the list structure was full.

CPExpert produces Rule CIC425 when the number of Structure Full conditions is greater than the value specified by the **CFNOSPCE** guidance variable in USOURCE(CICGUIDE). The default value for the **CFNOSPCE** is 0, indicating that CPExpert should produce Rule CIC425 whenever any Structure Full conditions occurred.

You can alter this analysis using the **CFNOSPCE** guidance variable. For example, if you wish Rule CIC324 produced only when more than one Structure Full condition occurred, specify:

```
%LET CFNOSPCE = 1 ; * TIMES LIST STRUCTURE WAS OUT OF SPACE;
```

Unless you have a very unique situation, you should **not** alter the CFNOSPCE guidance variable. You should always be aware of Structure Full situations.

Chapter 7.1.3: Percent data elements in use - CFPCTELE variable

CICS CFDT support is designed to provide sharing of working data within a sysplex, while maintaining update integrity of the data. The working data is held in a coupling facility data table, which is contained in a *named pool* located in coupling facility list structure. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a list structure in a coupling facility.

IBM suggests that no more than 75% of the list structure be used, to minimize the risk of the structure becoming full, to avoid triggering low space warning messages, and to avoid additional activity required to alter entry to element ratios. However, the default ELEMENTWARN and ENTRYWARN warning parameters have a default value of **80**, which specify that warnings and automatic ALTER actions should be first triggered when 80% of the elements or entries are used.

CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used. CFDT pool server statistics for the coupling facility are available in MXG file CICCFS6D. CPExpert uses data in CICCFS6D to calculate the maximum percent of the structure list elements that had been used.

CPExpert produces Rule CIC420 when the maximum percent structure list elements used is more than the value specified by the **CFPCTELE** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTELE** is 70 indicating that CPExpert should produce Rule CIC420 whenever more than 70% of the list entries had been used.

You can alter this analysis using the **CFPCTELE** guidance variable. For example, if you wish Rule CIC420 produced only when more than 60% of the data elements had been used, specify:


```
%LET CFPCTELE = 60 ; * MAXIMUM PERCENT DATA ELEMENTS USED;
```

Chapter 7.1.4: Percent list entries in use - CFPCTENT variable

CICS CFDT support is designed to provide sharing of working data within a sysplex, while maintaining update integrity of the data. The working data is held in a coupling facility data table, which is contained in a *named pool* located in coupling facility list structure. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a list structure in a coupling facility.

IBM suggests that no more than 75% of the list structure be used, to minimize the risk of the structure becoming full, to avoid triggering low space warning messages, and to avoid additional activity required to alter entry to element ratios. However, the default ELEMENTWARN and ENTRYWARN warning parameters have a default value of **80**, which specify that warnings and automatic ALTER actions should be first triggered when 80% of the elements or entries are used.

CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used. CFDT pool server statistics for the coupling facility are available in MXG file CICCFS6D. CPExpert uses data in CICCFS6D to calculate the maximum percent of the structure list elements that had been used.

CPExpert produces Rule CIC421 when the maximum percent data entries used is more than the value specified by the **CFPCTENT** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTENT** is 70 indicating that CPExpert should produce Rule CIC421 whenever more than 70% of the data entries had been used.

You can alter this analysis using the **CFPCTENT** guidance variable. For example, if you wish Rule CIC421 produced only when more than 60% of the list entries had been used, specify:

```
%LET CFPCTENT = 60 ; * MAXIMUM PERCENT LIST ENTRIES USED;
```


Chapter 7.1.5: Percent record not found - CFPCTRNF variable

Applications access data in a CFDT using standard file control commands (read, write, delete, etc.). Records are placed into a CFDT in one of two ways: (1) records placed in the CFDT by the initial loading of the CFDT from the source data set (if a source data set is defined), and (2) new records written to the CFDT after the CFDT has been loaded.

Applications can reference the CFDT during initial loading. These references will produce a "record not found" condition if the references are to records outside the range of those already loaded into the data table. These conditions should be small and can normally be ignored.

The "record not found" condition should not occur after loading of the data table. Since the VSAM source data set is not available after the initial loading of the CFDT (if a source data set existed), any "record not found" condition for normally indicates (1) a coding or logic error with the application, (2) an attempt to reference a record that should be in the CFDT (but was not in the CFDT because, for example, the MAXNUMRECS had been reached or the record had been suppressed by one of the user exits mentioned above), or (3) a deliberate reference to determine whether the record exists.

CFDT list structure statistics for the coupling facility are available in MXG file CICCFS6D. The S6RSP3CT variable contains a count of the number of times CICS attempted to read a record from the CFDT, but was unable to satisfy the read request because the record was not in the CFDT. CPExpert calculates the percent of requests that encountered a "specified entry (table or item) was not found" condition.

CPExpert produces Rule CIC423 when the percent requests that resulted in a "entry (table or item) not found" condition is greater than the **CFPCTRNF** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTRNF** guidance variable is 0.1, indicating that CPExpert should produce Rule CIC423 when more than 0.1% of the requests resulted in a "entry (table or item) not found" condition.

You can alter this analysis using the **CFPCTRNF** guidance variable. For example, if you wish Rule CIC423 produced any file control requests to a CFDT encountered a "record not found" condition, specify:

```
%LET CFPCTNOE = 0 ; * PERCENT RECORDS NOT FOUND IN CFDT;
```


Chapter 7.3: Server buffer pool statistics

A Coupling Facility Data Table is assigned to a *coupling facility data table pool* in a coupling facility. A CFDT pool consists of an XES list structure on the coupling facility. Access to a CFDT by CICS transactions running in an AOR is through a *CFDT pool server* that supports a specific named CFDT pool. In this context, the CFDT pool server is similar to a File Owning Region (FOR) that would be used for a normal shared data table.

The CFDT pool server is started in its own region, by executing DFHCFMN. Various parameters are provided to DFHCFMN (POOLNAME, list structure parameters, lock wait parameters, tuning parameters, etc.) to allow tailoring of the data sharing server.

A CFDT pool server must be started on each MVS image for each CFDT pool defined in a coupling facility which can be accessed from that MVS image. The Coupling Facility Data Table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

CICS automatically connects to the server for a given CFDT pool the first time that any CFDT within that CFDT pool is referenced. All CFDT pool access is performed by cross-memory calls to the CFDT server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the CFDT server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool.

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. If there is still insufficient storage to satisfy the request, the request fails.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D.

Chapter 7.3.1: AXMPGANY requests failed after retry - CFANYRQS

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert produces Rule CIC441 when the number of storage requests in the AXMPGANY pool that failed after retry (variable S9ANYRQS[19]) is greater than the value specified by the **CFANYRQS** guidance variable in USOURCE(CICGUIDE). The default value for the **CFANYRQS** is 0, indicating that CPExpert should produce Rule CIC441 when any storage request in the AXMPGANY pool failed after retry.

You can alter this analysis using the **CFANYRQS** guidance variable. Since Rule CIC441 describes a situation that has a high impact on the performance of the CICS region, **you should not normally change the CFANYRQS guidance variable**. However, if you have very unusual circumstance and you wish Rule CIC441 produced only when more than five storage requests in the AXMPGANY pool failed after retry, specify:

```
%LET CFANYRQS = 5 ; * LOC=ANY REQUESTS FAILED AFTER RETRY;
```

Chapter 7.3.2: AXMPGLOW requests failed after retry - CFLOWRQS

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert produces Rule CIC444 when the number of storage requests in the AXMPGLOW pool that failed after retry (variable S9LOWRQS[20]) is greater than the value

[19] Please note that the *CICS Performance Guides* describe S9ANYRQF as failed requests and S9ANYRQS as requests to release storage. These descriptions are "reversed" initially, but IBM Hursley corrected the *CICS Performance Guide with document* SC34-6009-05. S9ANYRQF is the requests to "free" or release storage and S9ANYRQS is number of failed requests because of "short on storage".

[20] Please note that the *CICS Performance Guides* describe S9LOWRQF as failed requests and S9LOWRQS as requests to release storage. These descriptions are "reversed" initially, but IBM Hursley corrected the *CICS Performance Guide with document* SC34-6009-05. S9ANYRQF is the requests to "free" or release storage and S9ANYRQS is number of failed requests because of "short on storage".

specified by the **CFLOWRQS** guidance variable in USOURCE(CICGUIDE). The default value for the **CFLOWRQS** is 0, indicating that CPExpert should produce Rule CIC444 when any storage request in the AXMPGANY pool failed after retry.

You can alter this analysis using the **CFLOWRQS** guidance variable. Since Rule CIC441 describes a situation that has a high impact on the performance of the CICS region, **you should not normally change the CFLOWRQS guidance variable**. However, if you have very unusual circumstance and you wish Rule CIC444 produced only when more than five storage requests in the AXMPGLOW pool failed after retry, specify:

```
%LET CFLOWRQS = 5 ; * LOC=LOW REQUESTS FAILED AFTER RETRY;
```

Chapter 7.3.3: Low percent AXMPGANY free storage - CFPCTAMN

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert uses data in CICCFS9D to calculate the minimum percent of free storage in the AXMPGANY pool.

CPExpert produces Rule CIC442 when the percent free storage in the AXMPGANY pool is less than the value specified by the **CFPCTAMN** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTAMN** is 25, indicating that CPExpert should produce Rule CIC442 whenever less than 25% of storage in the AXMPGANY pool is free.

You can alter this analysis using the **CFPCTAMN** guidance variable. For example, if you wish Rule CIC442 produced only when less than 25% of storage in the AXMPGANY pool is free, specify:

```
%LET CFPCTAMN = 5 ; * LOC=ANY PERCENT MINIMUM FREE STORAGE;
```


Chapter 7.3.4: AXMPGANY requests retried - CFPCTARC

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics.

When requests in the AXMPGANY pool are retried, this means that all free storage in the AXMPGANY pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the "compress attempt" to reduce fragmentation in the AXMPGANY storage POOL will correspondingly increase.

CPExpert uses data in CICCFS9D to calculate the percent of storage requests in the AXMPGANY pool initially failed and were retried.

CPExpert produces Rule CIC440 when the percent of storage requests in the AXMPGANY pool initially failed and were retried is greater than the value specified by the **CFPCTARC** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTARC** is .1, indicating that CPExpert should produce Rule CIC440 whenever more than one tenth percent of the storage requests in the AXMPGANY pool initially failed and were retried.

You can alter this analysis using the **CFPCTARC** guidance variable. For example, if you wish Rule CIC440 produced only when more than five percent of storage requests in the AXMPGANY pool initially failed and were retried , specify:

```
%LET CFPCTARC = 5 ; * PERCENT LOC=ANY REQUEST FAILED AND RETRIED;
```

Chapter 7.3.5: Low percent AXMPGLOW free storage - CFPCTLMN

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert uses data in CICFS9D to calculate the minimum percent of free storage in the AXMPGLOW pool.

CPExpert produces Rule CIC445 when the percent free storage in the AXMPGLOW pool is less than the value specified by the **CFPCTLMN** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTLMN** is 25, indicating that CPExpert should produce Rule CIC445 whenever less than 25% of storage in the AXMPGLOW pool is free.

You can alter this analysis using the **CFPCTLMN** guidance variable. For example, if you wish Rule CIC445 produced only when less than 25% of storage in the AXMPGLOW pool is free, specify:

```
%LET CFPCTLMN = 5 ; * LOC=LOW PERCENT MINIMUM FREE STORAGE;
```

Chapter 7.3.6: AXMPGANY requests retried - CFPCTLRC

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics.

When requests in the AXMPGLOW pool are retried, this means that all free storage in the AXMPGLOW pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the “compress attempt” to reduce fragmentation in the AXMPGLOW storage POOL will correspondingly increase.

CPExpert uses data in CICCFS9D to calculate the percent of storage requests in the AXMPGLOW pool initially failed and were retried. CPExpert produces Rule CIC443 when the percent of storage requests in the AXMPGLOW pool initially failed and were retried is greater than the value specified by the **CFPCTLRC** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTLRC** is .1, indicating that CPExpert should produce Rule CIC443 whenever more than one tenth percent of the storage requests in the AXMPGLOW pool initially failed and were retried.

You can alter this analysis using the **CFPCTLRC** guidance variable. For example, if you wish Rule CIC443 produced only when more than five percent of storage requests in the AXMPGLOW pool initially failed and were retried , specify:

```
%LET CFPCTLRC = 5 ; * PERCENT LOC=LOW REQUEST FAILED AND RETRIED;
```


Chapter 7.4 Specifying guidance for specific shared data tables

The guidance variables for shared data tables (including Coupling Facility Data Tables) are globally applied during CPExpert's analysis of shared data table performance. These global guidance variables might not be applicable to some shared data tables or Coupling Facility Data Tables, however. Guidance can **optionally** be applied to specific shared data tables or to specific Coupling Facility Data Table (CFDT) servers (and to the coupling facility structures used by the CFDT servers.

Guidance for specific shared data tables (or for CFDT servers) is accomplished by specifying **%LET DTGUIDE=Y;** in USOURCE(CICGUIDE), identifying the shared data table names (or CFDT server names) to which the guidance applies, and specifying guidance variables for the shared data tables or CFDT servers.

Please note that the guidance is specified for specific shared data tables (or CFDT servers). The guidance is NOT specified for specific CICS regions, nor may guidance for shared data tables be "overridden" on a region-by-region basis. This is because shared data tables are not associated with a specific CICS region (even though a CICS region will act as a File Owning Region for the shared data table), and because a CFDT server is not associated with a CICS region.

Exhibit 3-4 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the analysis guidance variables for specific shared data tables or CFDT server. As illustrated in Exhibit 3-4, guidance is specified **inside** the SAS macro comment statements (/* and */). The SAS macro comment statements may not be altered, as they control CPExpert's processing of the USOURCE(CICGUIDE) member.

The shared data tables (or CFDT servers) are identified by the TABLE statement, which is used to specify the shared data table (or CFDT server) to which the specific guidance applies.

Any number of shared data tables (or CFDT servers) may be defined with appropriate guidance specified for each tablel.

Following the TABLE statement are the individual guidance statements for the shared data table (or CFDT server) identified. Any of the global shared data table (or CFDT) guidance variables can be specified for the shared data table (or CFDT).

If a particular shared data table guidance variable statement is not present or has a null value, the global default will be used. **NOTE: a null value must be indicated by a semi-colon or a SAS error will result.**

The below example shows that specific guidance for the has been specified for the G1POOL2 shared temporary storage pool server. In this example, 10 was specified as the guidance for the TSPCTLRU variable.

```
%LET TSGUIDE = Y;        * GUIDANCE IS PROVIDED FOR INDIVIDUAL SHARED TS POOLS;
/* SPECIFY GUIDANCE FOR SHARED TEMPORARY STORAGE POOLS
POOL  = G1POOL2
TSPCTLRU  =10  * PERCENT LRU ACTIVITY         ;
*/
```

